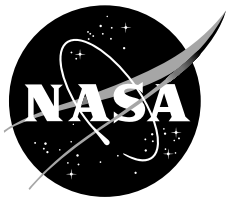
NASA/TM—2005-213455



A Program to Improve the Triangulated Surface Mesh Quality Along Aircraft Component Intersections

Susan E. Cliff
Ames Research Center
Moffett Field, California

May 2005

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

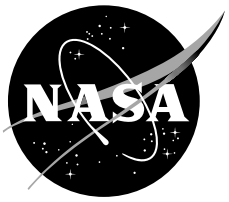
- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM—2005-213455



A Program to Improve the Triangulated Surface Mesh Quality Along Aircraft Component Intersections

Susan E. Cliff
Ames Research Center
Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

May 2005

Acknowledgments

The author is grateful to Scott D. Thomas for making more than 100 movies, and for his suggestions and help with many programming issues that arose during the development of this program. Without the existing software that Scott developed, this technique would neither have been thought possible nor have been realized. The author would also like to acknowledge that without the use of Mike Aftosmis's robust intersection program, this method would clearly not have been possible.

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

CONTENTS

A PROGRAM TO IMPROVE THE TRIANGULATED SURFACE MESH QUALITY ALONG AIRCRAFT COMPONENT INTERSECTIONS

Susan E. Cliff

Ames Research Center

SUMMARY

A computer program has been developed for improving the quality of unstructured triangulated surface meshes in the vicinity of component intersections. The method relies solely on point removal and edge swapping for improving the triangulations. It can be applied to any lifting surface component such as a wing, canard, or horizontal tail component intersected with a fuselage, or it can be applied to a pylon that is intersected with a wing, fuselage, or nacelle. The lifting surfaces or pylon are assumed to be aligned in the axial direction with closed trailing edges. The method currently maintains salient edges only at leading and trailing edges of the wing or pylon component. This method should work well for any shape of fuselage that is free of salient edges at the intersection. The method has been successfully demonstrated on a total of 125 different test cases that include both blunt and sharp wing leading edges. The code is targeted for use in the automated environment of numerical optimization where geometric perturbations to individual components can be critical to the aerodynamic performance of a vehicle. Histograms of triangle aspect ratios are reported to assess the quality of the triangles attached to the intersection curves before and after application of the program. Large improvements to the quality of the triangulations were obtained for the 125 test cases; the quality was sufficient for use with an automated tetrahedral mesh generation program that is used as part of an aerodynamic shape optimization method.

INTRODUCTION

Individual aircraft component surfaces are often represented by an unstructured mesh of triangles. Surface grids obtained through computer-aided design (CAD) systems or developed for detailed or preliminary design that extend through components to which they are attached simplify the mesh-generation process and allow for movement of the individual components. CAD-based and external intersection methods exist that intersect and remove the interior triangles from the joined components. Although accurate surface representation is achieved, poor quality triangles are nearly always formed along the intersection. The quality of the triangles along the intersection is inadequate for use with tetrahedral volume mesh generators.

Tetrahedral mesh-generation methods require moderately high-quality surface triangulations to generate high-quality tetrahedral volume meshes since each of the surface faces forms at least one of the faces of a tetrahedron. Triangle aspect ratio, which is defined as the inscribed circle radius (r) divided by the circumscribed circle radius (R), is used to measure surface mesh quality (the highest-

quality triangle is equilateral, aspect ratio 1/2). Aerodynamic shape optimization (ASO) methods employing tetrahedral meshes have been limited to design shape changes that do not change the intersection between components (ref. 1). Shape changes are applied either to all components (such as fuselage camber being carried onto the wing component), or the geometric changes are applied to individual components outside the intersected region. This limitation may significantly reduce the aerodynamic performance gain that could be achieved through shape optimization of a particular vehicle. An automated method to improve the quality of triangulations in the region of intersected components is the subject of this report. The method is referred to as the intersection clean up (ICU) method. The ICU method uses an existing surface triangulation intersection program that was developed for Cartesian meshes by M. Aftosmis (ref. 2). This method is very robust, and it maintains all points outside the intersection to maintain the shape of the individual components and removes all interior points. However, the intersection method creates poor-quality triangulations (triangles with small aspect ratios) and can create degenerates such as co-linear or nearly co-linear triangular faces along the intersection regions. (The Cartesian method is insensitive to poor quality or degenerate triangulations, whereas the tetrahedral grid method is dependent upon a reasonable-quality surface grid.)

Good-quality surface triangulations are important for use in advanced ASO methods. Unstructured grid-based methods are ideal for handling geometries that have newly formed intersection regions. Quality surface meshes allow for the use of a robust tetrahedral-grid-based Euler method known as AIRPLANE (ref. 3) to be used for ASO. This optimization method has been used to optimize complex vehicles with short grid-generation lead times almost regardless of geometric complexity. The use of this method is further warranted since it has recently been developed with a computationally efficient method of achieving gradient computations, known as an Adjoint method (ref. 4). Formerly, the method relied solely on finite differences for its gradient computations (ref. 1).

The ICU method should be useful to other methods that use triangulated surfaces that require the removal of nonphysical triangulations (zero-area triangles) such as a panel method used at Ames named CBAERO (ref. 5).

INTERSECTION CLEAN-UP ALGORITHM DESCRIPTION

The ICU program is a multistepped procedure. It comprises four steps that are repeated three times. The input requirements are two closed (water-tight) surfaces and a non-closed surface grid with a single opening. Examples of these surfaces are shown in figure 1 for a wing, body, and empennage with a horizontal T-tail, and vertical tail surfaces. Typically the wing with an open root section is used as the surface grid of the non-closed surface (fig. 1(a)). The edges around the opening of the non-closed surface are identified and used to determine the “distribution curve”—the basis of the point distribution on the final intersection. This will allow ICU to maintain the clustering of points at the leading or trailing edges or to preserve a clustering on either the upper or lower surface. A sectional input could be used alternatively, if an open surface grid is not available. The water-tight components of the configuration must intersect in only one location, thus the wing root must be entirely inside the fuselage and cannot be closed off at the symmetry plane if the fuselage is closed at the symmetry plane (fig. 1(b)). The wing must be the first component of the two water-tight

surfaces, and the fuselage and any remaining surfaces are considered to be the second component. Figure 1(c) shows that the wing root is a closed surface, since it is behind the fuselage in figure 1(b).

The ICU program removes triangles, and/or swaps edges along the intersection. This causes a less-accurate representation of the surface than was represented by the original intersection. This is a necessary consequence of the program to improve the quality of the triangulation while keeping much of the surface triangulation unchanged outside the intersection. The accuracy with which a triangulated surface mesh represents a surface is only as good as the density of points that are provided in the original surface. It is the responsibility of the user to provide sufficient point densities on the original meshes of individual aircraft components, and to provide an adequate distribution curve. If a proper distribution curve is provided and the original surface meshes on the individual components have sufficient point densities to accurately represent the true surfaces, then this method should not degrade the accuracy of the surface representation.

The surface grids of two water-tight components are intersected via an external call to M. Aftosmis's INTERSECT program. INTERSECT provides the user with the union of the two components with the interior points removed. The triangles associated with each of the two components are retained. Retention of the component's numbering allows the ICU program to find the intersection by obtaining the single-use edges on either component. The intersection points are arranged to form a polygonal curve of either clockwise or counterclockwise orientation beginning at the trailing edge upper or lower surface. This process is currently done through an external call to a program named APTRIANG, which was written to the author's guidelines by S. D. Thomas. This program is called before each step in ICU to obtain new intersections and to remove collapsed triangles. Each selected poor-quality triangle is collapsed during the first three steps of the ICU program by mapping one triangle point to one of the other two points. APTRIANG can detect this particular type of triangle collapse and it can remove the collapsed face from the database. APTRIANG is also used to orient all the faces of a dataset so that all surface normals are pointing in same directions (in or out). It is also used to remove duplicate points or points that are within a specified distance of other points in the triangulation. Occasionally INTERSECT results in ill-formed triangulations with points very close together where an intersection point of one component does not match to the intersection point in the other component. This is easily handled with the aforementioned feature in APTRIANG. APTRIANG has many other features, but those are the primary ones used by external calls from ICU.

The following paragraphs give a brief description of each of the four procedural steps in the ICU program. Programming details are omitted with regard to maintaining the leading and trailing edge points and maintaining the shape of the salient edges when sharp-edged wings are used. This method was also designed to improve the quality of intersections of wings with blunt leading edges without having to specify the shape of the leading edge. Obtaining good triangulations around both sharp and blunt leading edges was the main challenge of creating this program.

Step 1 of the ICU Program

The first step of the program is to remove points or groups of points on the two components that are too close to each other. Groups of 1, 2, 3, or 4 neighboring points on the intersection curve are selected for removal by comparison with the spacing of the distribution curve with the spacing of the intersection curve. The distribution and intersection curves are normalized by the length of the chord in the streamwise direction (axially), thus the two curves are parameterized between values of 0 to 1, and spacing differences can be compared. If the spacing on the intersection between points after removal of the single point (or multiple points) is less than a specified fraction of the spacing of the distribution curve, then the point (or points) can be removed. A single intersection point is a candidate for removal only if the point and its two neighboring points all have a common third point off the intersection on both components. This is demonstrated in figure 2. Only the red edges are candidates for removal. This is a fairly simple step because removing these points will not cause any crossed-over triangles (this is discussed during step 2 of the procedure). The points are removed by assigning the candidate point to one of its neighbor's point values on the intersection; it does not matter which neighbor (to the right or left) in this single-point removal case.

If two neighbor points are found as candidates for removal (fig. 3), then either one or both points are removed, depending on the intersection-point spacing after single or multiple points are removed. ICU first attempts to remove both points if the spacing after removal is less than the reference curve spacing; otherwise it attempts to remove one or the other of the two candidate points. The points chosen for removal are assigned the value of the last non-chosen point.

The situation whereby three candidate points are labeled for removal is shown in figure 4. The algorithm attempts to remove all three points, and then groups of two, and then single points. The algorithm checks for groups of up to four candidate points for removal and follows a similar algorithm. The fact that there could be more than a group of four points for removal is handled by the algorithm repeating this step two additional times.

A test case of a truncated wing that is intersected with a truncated cylindrically shaped fuselage is used to show the results of each step of the ICU program. By truncating the components, the turnaround time was greatly improved during the development of this code. This case consisted of 3,094 surface points, and the wall-clock time for ICU was about 268 seconds on a 225-MHz R10000 chip SGI workstation. This test case was used as the primary dataset to debug the code. In an attempt to encounter many different triangulations and situations or combinations of circumstances as one would encounter during the course of optimization, the wing was shifted in the vertical direction to 40 additional positions. Figures 5(a) and 5(b) show the range of wing positions, with the wing in the highest vertical position in figure 5(a) and lowest position in figure 5(b). Also tested were 38 additional positions equally spaced between the two positions shown in figures 5(a) and 5(b). The triangles on the wing are red, and those associated with the fuselage are blue. The tip of the wing, the symmetry plane, and the forward and aft surfaces of the fuselage are removed in these figures to inspect the inside of the geometry. The intersection between the wing and fuselage is shown in detail in subsequent figures as the grid is modified via the ICU program for the region depicted by the white box in figure 6. The nonshifted case was selected for the examples used to show the improvements of the triangulations during the execution of the ICU program.

The original wing intersection and the results after step 1 of the ICU program are shown in figures 7(a) and (b) for the region depicted in figure 6. Figure 7(a) is the intersection before clean-up, and figure 7(b) is after step 1 of the ICU program.

Step 2 of the ICU Program

Step 2 of the program is similar to step 1, in that points along the intersection are removed if they are too close to each other. However, this step does not require that the intersection points have a common third point with neighboring intersection points. The program marches along the intersection in either a clockwise or counterclockwise manner, through a loop with index i. The algorithm attempts to replace Cartesian coordinates at i + 1 with those at i. This has the effect of moving point i + 1 to i. The resulting triangle after collapse will be i, i + 2, and the third point of the triangle of the original i + 1, and i + 2 intersection points ($P3(i + 1,i + 2)$). In general this step of the program cannot be done without checking prior to collapse whether the collapse will create crossed-over triangles. The nomenclature used previously, $P3(P1,P2)$, is used to mean the third point ($P3$) of the triangle with vertices $P1$ and $P2$ throughout the text of this report.

In order to avoid wrinkles or crossed-over triangles, the angles of the triangles formed with the two points on the intersection and the point on the surface of each component must be computed for the original triangulation, and also for the new triangulation after edge collapse. The interior angles are computed using the law of cosines. The angles at vertex i of the original triangulation and the resulting triangulation after edge collapse are compared. Rather than compute the angles for the newly formed triangle by (i, i + 2) and the original $P3(i + 1, i + 2)$ intersection points, the angle between i, i + 1, and $P3(i + 1, i + 2)$ is computed. This is a fictitious triangle that is never formed, but used for the computation of β(new) so that this approach will work for nonstraight intersections. β is the angle on side i of each triangle and is the angle between i + 1, i, and $P3(i, i + 1)$, the triangle off the intersection, and β(new) is the angle measurement between i + 1, i and $P3(i + 1, i + 2)$ intersection points. β and β(new) are compared, and if β(new) is less than or equal to β, then it is possible to collapse the point i + 1 to i. Figure 8(a) shows an example of when the cross-over test shows that no cross-over will occur. Figure 8(b) shows an example where a cross-over would occur, and thus the intersection must not be altered. Note that in figures 8(a) and 8(b), only one component is shown to simplify the drawing, but both components would have to have no cross-overs and the spacing would have to be sufficiently small to warrant a collapse. Also note that the direction in which i increases is different in figures 8(a) and 8(b) since the direction is irrelevant to the procedure. The spacing criterion for edge collapse is that the normalized edge length between i and i + 1 be less than 1/3 of the normalized reference curve spacing. This value can be increased for a more aggressive clean-up. This entire step is repeated until no edge collapses are obtained. Thus, surface grid files are written after each pass around the intersection, and APTRIANG is used to remove the duplicate points/edges of the collapsed triangles before the method is reapplied. The step is complete when no edge collapses are made in the previous pass.

The resulting intersection after step 2 of the ICU program is shown in figure 9. Compare this figure with figure 7(b) to see the improvement in the surface mesh quality.

Step 3 of the ICU Program

The third step in the program is used to collapse edges/remove points off the intersection; that is, drive points from the individual components to the intersection curve. This differs from the first two steps, which were collapsing edges on the intersection curve. Simply stated, this step in the program is used to remove points on the individual components that are too close to the intersection curve. It treats only triangles with third points between two successive intersection points. It looks at the α and β angles of each successive pair of points on the intersection and the pair's third point of the triangle that lies on each individual component. It treats the two components separately. β is the included angle formed with the intersection point i, and α is the angle formed with the next successive intersection point i + 1. This is shown in figure 10. The length *a* is the distance from point i to the third point of the triangle, and *b* is the length from i + 1 to the third triangle point. This step collapses edges based on two criteria, the size of the angles α and β and the lengths of the edges *a* and *b*. The top and middle diagrams show examples of where the angular measurements of β and α are small, respectively. If α or β are less than 20 degrees (arbitrarily chosen—larger angles would allow for more aggressive clean-up), then the triangle will be collapsed. As depicted in the diagrams, it replaces the third point of the triangle (not on the intersection) with one of the other two triangle points, the closer of the two points that lie on the intersection curve. This step also collapses based on the normalized edge lengths (edge lengths *a* and *b* divided by the chord). It compares the normalized edge length with the normalized reference curve spacing. If these normalized dimensions are less than 1/5 of the normalized reference curve spacing, then it attempts to collapse to the shorter edge. This takes care of some of the points that were too close together in step 2, but could not be collapsed because of cross-over as depicted in figure 8(b).

There are several instances whereby step 3 cannot be implemented based on just the angle and edge-length criteria. The problem is that the method can produce degenerate triangulations with nearly co-linear points; these triangulations are referred to as pancakes. The triangulations upstream and downstream of the current candidate triangle for collapse (with vertices i, i + 1, *P*3(i, i + 1)) must be examined prior to edge collapse. The upstream points are labeled *P*3 + 1, *P*3 + 2, etc. and the downstream points are labeled *P*3 – 1, *P*3 – 2, etc. The region that is examined is depicted in figure 11. This figure shows a situation that the method would cause a pancake if *P*3 were replaced with the point at i + 1, since *P*3 is equal to *P*3 – 1, and *b* is less than *a*. A similar situation occurs if *P*3 is equal to *P*3 + 1 and if *a* were less than *b*. A second example of a non-collapsible state is shown in figure 12. Here *P*3 is equal to *P*3 – 2, and a collapse is not possible to either i or i + 1. Similarly, if *P*3 is equal to *P*3 + 2, *P*3 cannot be replaced with either i or i + 1 without creating a pancake. The third example of a non-collapsible triangle involves the marking of the new third point, *P*3(new), of the last collapsed triangle. In figure 13, if *b* is less than *a* and *P*3 is replaced with the intersection point i + 1, then the new third point after the collapse is the alternate third point of the triangle with the common dual-use edge (*P*3, i + 1) that is not the intersection point i. In this case, *P*3(new) is equal to *P*3 + 1, and *P*3 + 1 cannot be replaced. The fourth example also involves marking points based on the situation depicted in figure 14. Here it is acceptable to collapse side *b (P*3 and *P*3 + 1) to the intersection point i + 1. If the third point of the dual-use edge (with vertices (*P*3, i + 1) does not equal i, but equals i + 2, then you must mark the dual-use edge's third triangle point for edge (*P*3,i + 2), that does not equal i + 1, as nonreplaceable. In this case, the nonreplaceable point is *P*3 + 2, circled in red in the figure. It is not necessarily *P*3 + 2, but is the third point of the triangle with edge (*P*3, i + 2) that is not i + 1. In the reverse situation (not depicted), whereby side *a* is col-

lapsed and the dual-use edge ($P3$, i) is equal to i – 1, not i + 1, the noncollapsible point is the third point of the triangle with common edge ($P3$, i – 1) that is not equal to i.

Improvements to the test case after step 3 of ICU are shown in figure 15 (compare with fig. 9). It has collapsed two triangles on the wing and one triangle on the body. Step 3 is skipped when it encounters one of the situations outlined previously that cause a degenerate pancake triangulation. The thin triangles that could not be collapsed through this step are repaired through re-application of this step with different neighboring triangles from the cycling, through the other steps of this program, or through the use of edge swapping (step 4 of program).

Step 4 of the ICU Program

The fourth step of the program is used to improve the quality of the triangulation through edge swapping. Unlike the former step, it handles surface points that are close to the intersection that are not between two intersection points. Four types of swaps are used in ICU. The first two are referred to as $P4$ and $P5$ swaps, and are shown in figure 16. The two edges joining the intersection points i and i + 1 and $P3$(i, i + 1) have edge lengths a and b, respectively. The point $P4$ is the third point of the triangle with the dual-use edge a that is not i + 1. $P5$ is the third point of the triangle with the dual-use edge b that is not i. The candidate swaps are edge a swapped with edge (i + 1, $P4$) and with edge b swapped with (i, $P5$). The candidate swaps are shown as red dashed lines. The aspect ratios are compared for the two original triangles and the two triangles formed after a swap. The computation of the aspect ratio, the in-radius (r) to circumcircle ratio (R), require only the edge lengths of each triangle. Poor-quality triangles have values near zero, and equilateral triangles (greatest quality) have a value of 1/2. The minimum ratio of the original two triangles and minimum ratio of the two newly formed triangles are compared. A swap is made if the minimum ratio of the two swapped triangles is greater than the minimum ratio of the two original triangles. In the figure, a $P4$ swap would clearly result in an improvement to the triangulation.

It should be noted that before a swap is ever examined, it has to be determined that a swap could physically occur without cross-over. This requires that the included angles of the triangles be computed in addition to the edge lengths. For example, for a $P4$ swap to be physically possible, the angle at i formed with i + 1, and $P4$ must be less than 180 degrees, and the angles that form the angle (i + 1, $P3$, $P4$) must also be less than 180 degrees. Each type of swap has its own two angle tests that have to be satisfied before a swap is examined or attempted.

The $P4$ and $P5$ swaps were the first two types of swaps that were coded, but it was soon realized that additional swaps were required. A situation arose where the points $P3$ and $P5$ were spread apart, forming a poor-quality triangle with the points $P3$, $P5$, i + 1. The situation is shown in figure 17. $P6$ is the third point of the triangle with edge ($P3$, $P5$) that is not equal to i + 1. In the figure, swapping the edge ($P3$, $P5$) with edge (i + 1, $P6$) improves the quality of the triangulation. $P6$ swaps are not allowed in the leading and trailing edge regions of the wing component to maintain the sharp edges.

The last type of swap that is used is a $P7$ swap, shown in Figure 18. $P7$ is found by finding the third point of the triangle with edge ($P3$, $P4$) that is not equal to i. This type of swap is also not allowed in the leading and trailing edge region of the wing component in order to maintain the sharp edge. The

*P*7 swap was necessary to improve the triangulations on the body component in the trailing edge region.

Only one swap is permitted for each i as the index traverses the intersection points. It checks whether *P*4, *P*5, *P*6, and *P*7 swaps are beneficial, respectively. It appears that if one of these types of swaps is beneficial, the other swap types would not be beneficial. Clearly, it is never the case that all swaps would ever be necessary. The program cycles through step 4 three times, thus at most three of the four types of swaps could be obtained for each edge along the intersection. It is doubtful that more than two would ever be needed.

Prior to performing any of the four types of edge swaps, the program checks for degenerate triangles. It makes the assumption that the point representation is sufficiently dense that any three points along the intersection are nearly co-linear (except for the leading and trailing edge points). Thus, pancakes can be found by checking for an edge joining i to i + 2. It handles this situation by swapping edge (i, i + 2). This edge is a dual-use edge. It finds the third point of the triangle with the common edge that is not i + 1, and swaps the edge with this point and i + 1 with the edge (i, i + 2). A situation also arose where a three-edge pancake was found; an edge joining i to i + 3 was encountered. This is also handled with a swapping technique similar to the aforementioned.

The results of this edge-swapping routine for the test case are shown in figure 19 (compare with fig. 15). Note that three swaps were performed on the wing, and four on the body. Swaps on both the wing and body are partially visible on the left side of the figure. The triangulation looks very good, but on the wing component (about 1/4 of a frame to the left of the right side of the frame) is a point that is too close to the intersection with an included angle of less than 20 degrees. The result of the swapping has created a triangle that can be repaired via step 3 of ICU. The next two steps of the program, steps 5 and 6, do not alter the triangulation in the displayed region. Step 5 repeats step 1, and step 6 repeats step 2. Step 7 repeats step 3, which collapses the aforementioned triangle on the wing (fig. 20). Steps 8–12 did not alter the triangulation in this region, so results are not shown. The resulting grid (fig. 20) is greatly improved compared with the original triangulation in this region (Fig. 7(a)).

In addition to visual inspection, the aspect ratios (the in-radius (r)/circumradius ratios (R)) were computed for the triangles along the intersection (triangles with two of the three points on the intersection) before and after application of ICU. The ratio was multiplied by 2 for a ratio range from 0 to 1. The number of triangles with ratio values between each tenth were counted to separate the triangles into 10 classes to show the frequency (number) of the triangles within each class interval before and after applying ICU (fig. 21). Percentages are reported rather than the actual frequency values since the number of triangles along component intersection are significantly reduced after application of ICU. This shows that the largest percentage of triangles were of the poorest quality ($0 < 2r/R < 0.1$) before application of ICU, and none of the triangles were of the poorest quality after application of ICU. Furthermore, the largest percentage of triangles were of the best quality ($0.9 < 2r/R < 1.0$) after application of ICU. The minimum, maximum, and average aspect ratios (2r/R) were 0.0009, 0.999, and 0.387, respectively, before ICU was applied. After application of ICU, the minimum, maximum, and average aspect ratios were 0.193, 0.999, and 0.737, respectively. The average ratio after application of ICU represents an improvement of 90% compared with the triangulation before application of ICU. The improvement over the minimum quality triangle is 21,344%.

ICU TEST CASE RESULTS

The program was further demonstrated on a total of 125 test cases using two different configurations. The intersections on both configurations lie between wing and body components. The first test case, already introduced in the algorithm description, has a sharp leading edge wing component, and the second test case has a blunt wing leading edge (typical supercritical wing shape). The sharp-leading-edged wing test database was translated vertically to 41 different positions. The blunt-edged wing test case was tested with four levels of mesh densities. For each level, the wing was translated vertically to 21 different positions, for a total of 84 test cases with blunt leading edges. Thus, both blunt and sharp test cases comprise 125 test cases. The results of the method were assessed by visual inspection of triangles formed along the intersections at all 12 steps of the program during the development of the algorithm. Movies were made at several locations (seven locations for the sharp-leading-edge test cases and four locations for the blunt-leading-edge test cases) along the intersections of all test cases so that individual frames could be manually advanced at each inspection location to show the results of each step of the ICU process at all intersections. Thus, 10,439 (((4 x 4 x 41) + (7 x 21)) x 13) frames have been carefully inspected for algorithm programming errors. The movie-making process was automated, but took 1–2 days for all the movie frames to be made. The inspection time took an hour or two to review all the frames. The use of movies was vital to the development of the code. Additional movies were made at locations where programming errors were encountered and in most cases they revealed the step in the program where the program was flawed.

Sharp-Leading-Edge Wing Test Results

Three of the seven views that were used for visual inspection during the development of ICU for the sharp-wing test case are shown for two of the 41 vertical wing locations in figures 22 and 23. These views include the wing trailing and leading edge as viewed from above, and a mid-wing location viewed from below shown in the top, middle, and bottom portion of the figures, respectively. These two vertical locations of the wing were randomly selected from the 41 positions since only a small number of cases can be presented in this report and since all cases had nearly equivalent mesh quality improvements. As demonstrated in the figures, the wing maintains its sharp leading and trailing edges. This was accomplished on all the test cases. Maintaining these two salient edges required modifications to the program to properly maintain the sharp wing leading and trailing edges. Many changes were required in all four steps of the program to achieve this, but these details will not be reported here since it is beyond the scope of this report.

The mesh quality statistics in terms of the percentages of triangles along the intersection within each of the ten class intervals are shown in figures 24 and 25 for the two translations. The percentages of all 41 test cases were averaged and are shown in figure 26. The percentage of triangles in the poorer-quality categories is very low (less than 5% for the first four categories), and the percentage of triangles increases nearly linearly to the higher-quality categories, reaching approximately 23% of the triangles in the highest-quality category. The similarity with the randomly selected individual results of the two test cases and the average results for all 41 test cases indicate that ICU performed well for all 41 test cases. The minimum, maximum, and average aspect ratios (2r/R) were also aver-

aged for all 41 test cases. These average values before application of ICU were 0.0007, 0.998, and 0.407, respectively; after application of ICU the averages were 0.164, 0.999, and 0.730, respectively.

The final and most important measurement of success for the ICU program is determined by the success of the automated tetrahedral volume mesh generator (ref. 6) that is used within the optimization method used at Ames. Hence, the mesh generator was attempted on all the intersections after clean-up, and was successful in generating volume meshes for all 125 test cases. A few of the test cases were run through the AIRPLANE flow solver for additional confirmation of the success of the ICU program.

Blunt-Leading-Edge Test Results

The blunt-leading-edge test case had four levels of surface mesh refinement. For each grid refinement level the wing was vertically translated to 21 different positions, every inch from 0 to –20 inches before intersecting with the fuselage. Figures 27(a) and 27(b) show the wing in the uppermost vertical position (Shift = 0) and the lowest position (Shift = –20), respectively. The blunt-leading-edge test cases were at first flawed because of programming errors that did not arise until the blunt-leading-edge wing was attempted. After these were corrected, the program was successful at handling blunt leading edges and able to maintain the leading edge point of all the blunt-leading-edge test cases. The five inspection sites were evaluated via movies for each region of the intersection and found to also produce triangulations of moderate to high quality along the intersection after ICU was applied. Three of the five inspection sites are shown for one of the wing positions (the wing shifted by –8 inches) for each of the four levels of mesh refinement. The approximate location of the inspection regions are shown from afar with the same viewing angle as subsequent figures for the coarse mesh grids prior to the application of the ICU program (figs. 28(a)–(c)). These three inspection regions are shown for each level of mesh refinement in figures 29–32. Compare the left (pre-ICU) and right sides (post-ICU) of these figures. The upper two figures are of the wing trailing edge region viewed from above the wing; the middle two figures are of the wing leading edge also viewed from above the wing, and the bottom two figures are of the mid-chord of the wing viewed from above the wing. The coarse-mesh test case (fig. 29) shows faceting on the resulting triangulation as observed by brightness changes on the triangles upstream of the wing trailing edge on the fuselage component. This coarse mesh is used for multigridding purposes for the Euler flow solver, and is not of sufficient density to predict accurate flow solutions. This extreme example highlights the downfall of this method, in that the method lessens the accuracy of the surface representations in the region of the intersection. However, if adequate density surface meshes are used, as in the finer mesh examples (figs. 30–32), this effect is diminished to the point of being non-apparent and definitely non-problematic. The shaded surfaces in these finer-mesh results do not show any shading differences to indicate faceting of the surface. The wall-clock times for the four grid density levels (from coarse to fine) were 363, 691, 1495, and 4404 seconds for grids of 4,267, 10,831, 31,418, and 44,504 surface mesh points, respectively, on a SGI workstation with a 225-MHz R10000 chip.

The histograms of the average-aspect-ratio values within the ten quality categories are shown for the four mesh densities in figures 33–36. The pre- and post-ICU results are shown. These results also show good before-and-after improvements in cell quality along the intersection edge (compare figs. 33–36 with figs. 24–26). However, the blunt-leading-edge test cases do not obtain as high a percent-

age of triangles in the best-quality category. This is thought to be because the leading edge region on the blunt configuration was developed as a structured mesh to eliminate faceting. These triangles are of slightly poor quality, with one angle of each triangle being approximately 90 degrees, whereas a purely unstructured grid can achieve equilateral triangles.

The resulting surface meshes after application of ICU to all 84 blunt-leading-edge test cases were run through the tetrahedral volume grid generator. The success rate was 100%.

FUTURE WORK

The program will be coupled to the optimization framework at Ames and applied to an optimization problem where the wing is free to translate and geometric shape changes can be applied to the wing surfaces that extend inside the fuselage. The incorporation the ICU program to the optimization environment will allow for testing of hundreds of intersections. The program will subsequently be examined for computational speed improvements. Clearly, the subroutines of APTRIANG can be incorporated into the source code to eliminate the external calls and the reading and writing of the surface meshes. The code may be further sped up by truncating the components so that only surface triangles in the region of the intersection will be used with the ICU program. This will involve automating the truncation of components, forming new surface triangulations on truncation regions, and removing these temporary surface meshes after ICU is applied. The symmetry plane closure requirement could be achieved by using a pre-existing program “zip-up” (written by S. D. Thomas) that can automatically provide the symmetry plane closure instead of forcing the user to provide closed surface components.

In addition, the code may be extended to work for any intersection, in any orientation. Currently this demonstration code assumes that the wing leading edge is the most upstream point in the axial direction. Salient edges should be found automatically and maintained. Currently the leading and trailing edges are the only salient edges that are maintained. This would allow for an open-trailing-edge test case, with two salient edges to maintain, rather than the closed-trailing-edge case that is currently required.

REFERENCES

1. Cliff, S. E.; Thomas, S. D.; Baker, T. J.; Jameson, A.; and Hicks, R. M.: Aerodynamic Shape Optimization Using Unstructured Grid Method. AIAA Paper No. 02-5550, presented at the 9th Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Ga., Sept. 2002.

2. Aftosmis, M. J.; Berger, M. J.; and Melton, J. E.: Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry. AIAA Paper No. 97-0196.

3. Jameson, Antony; and Baker, Timothy J.: Improvements to the Aircraft Euler Method. AIAA Paper 87-0353, presented at the 25th AIAA Aerospace Sciences Meeting, Reno, Nev., Jan. 1987.

4. Jameson, Antony; Shankaran, Sriram; Martinelli, Luigi; Cliff, Susan; and Thomas, Scott: Aerodynamic Shape Optimization of Supersonic Aircraft Configurations. AIAA Paper No. 2005-1013, presented at the 43rd AIAA Aerospace Sciences Meeting, Reno, Nev., Jan. 2005.

5. Kinney, D.: Aero-Thermodynamics for Conceptual Design. AIAA-2004-31, presented at the 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nev., Jan. 5–8, 2004.

6. Baker, T. J.; and Vassberg, J. C.: Tetrahedral Mesh Generation and Optimization. Proceedings of the 6th International Conference on Numerical Grid Generation, International Society of Grid Generation (ISGG), Greenwich, U.K., 1998, pp. 337–349.

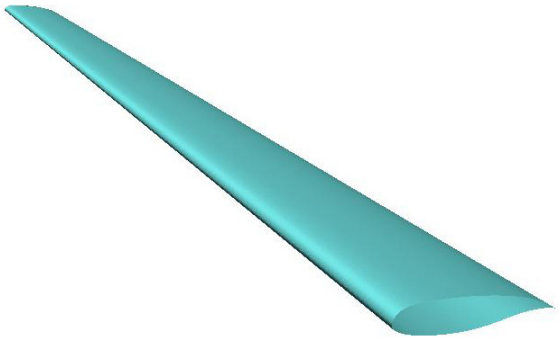


Figure 1(a). Wing surface grid with an open root section. Distribution of points on the root section is the basis of the point distributions along the intersections.

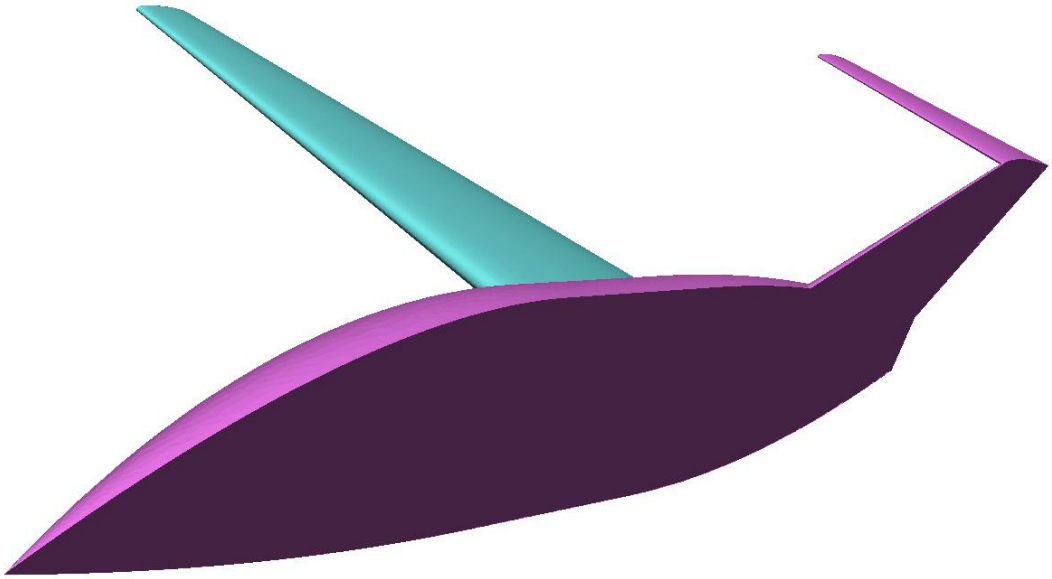


Figure 1(b). Wing and body closed water-tight surface grids.

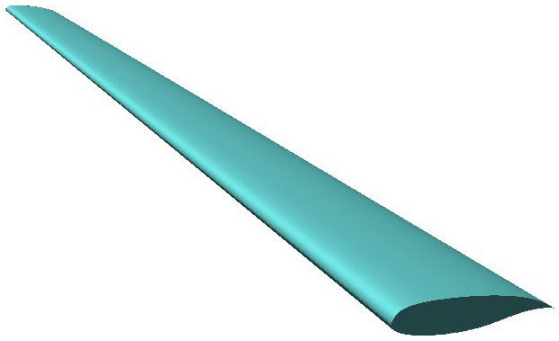


Figure 1(c). View of wing surface grid with closed "water-tight" surface with fuselage and tails removed.

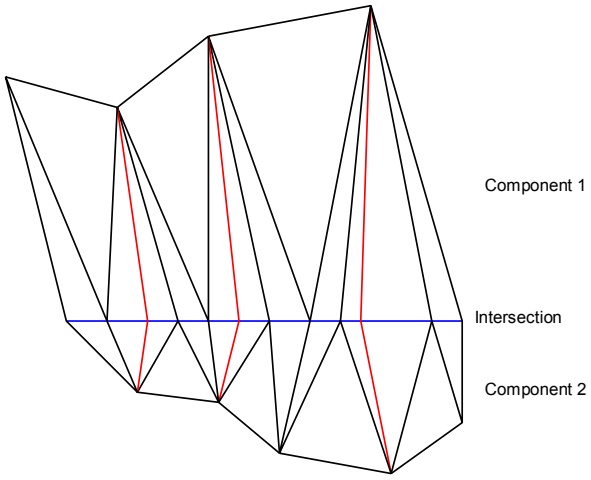


Figure 2. Single candidate points for removal along intersection are depicted (step 1 of ICU).

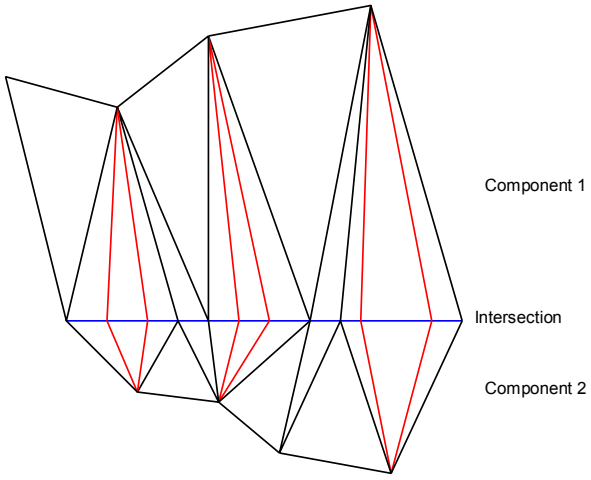


Figure 3. Groups of two candidate points for removal along intersection are depicted (step 1 of ICU).

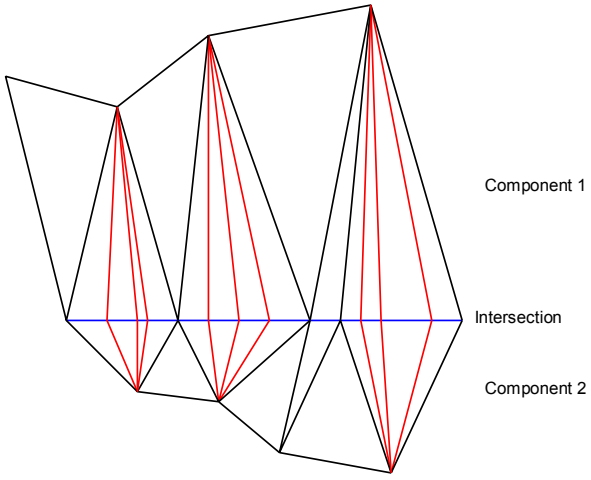


Figure 4. Groups of three candidate points for removal along intersection are depicted (step 1 of ICU).

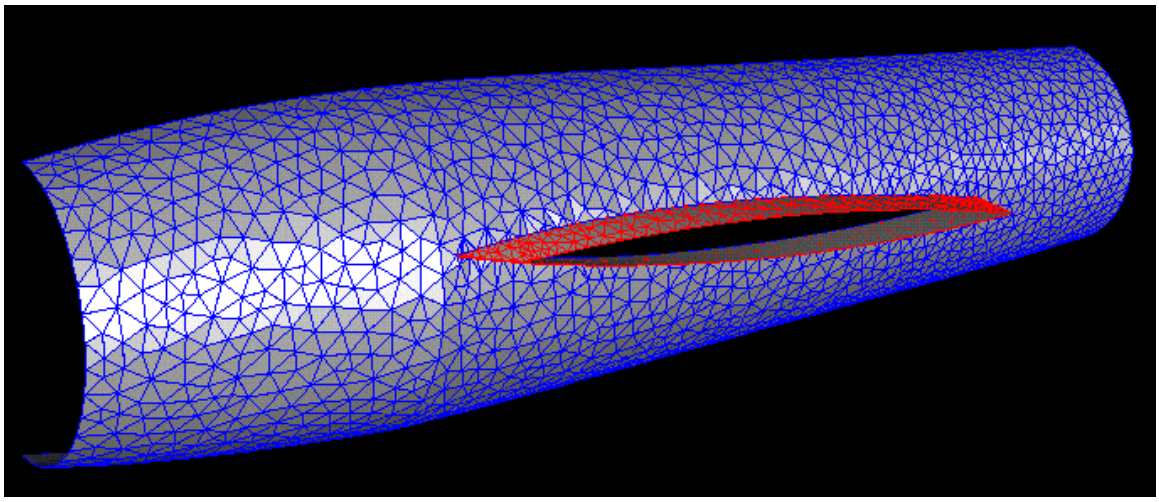


Figure 5(a). Test configuration with sharp leading edge, wing shifted to the highest position, wing shifted 2.6 inches from its original position.

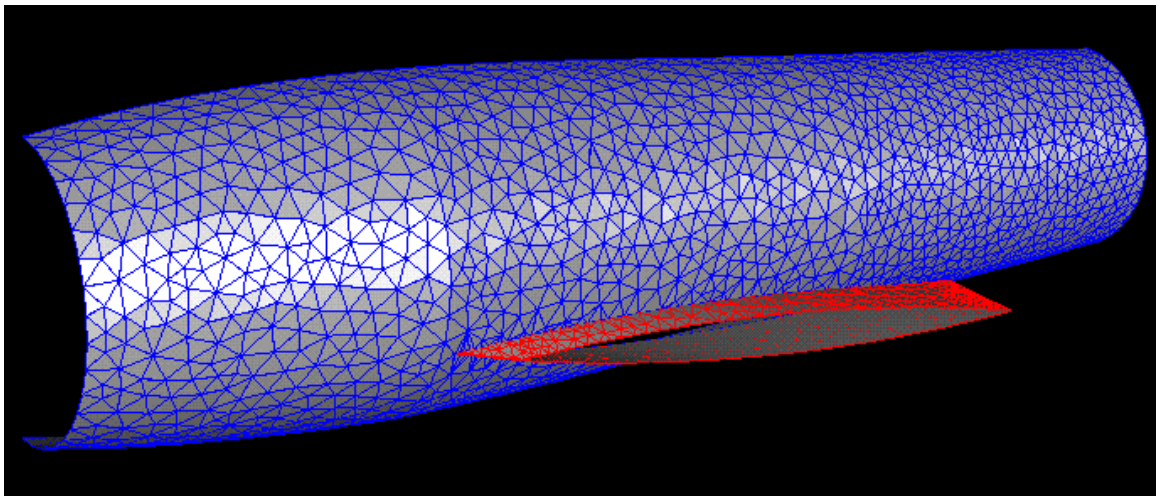


Figure 5(b). Test configuration with sharp leading edge, wing shifted to the lowest position, wing shifted –1.4 inches from its original position.

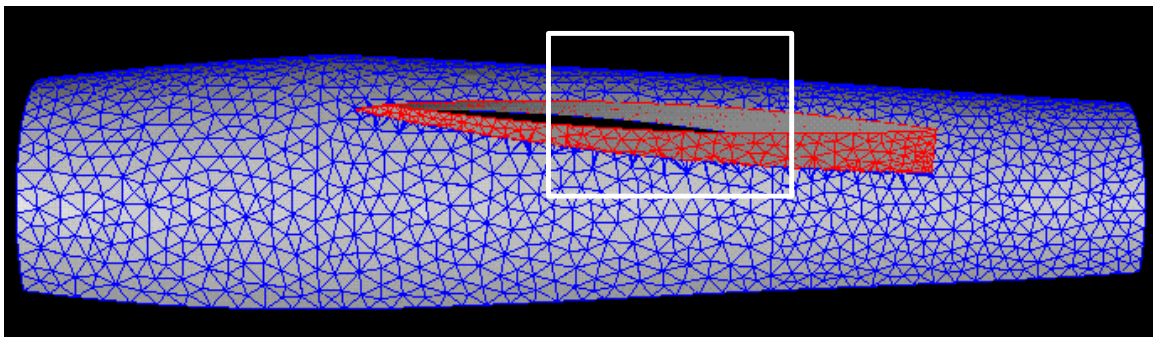


Figure 6. Inspection area of the test configuration with the wing in its original position is inside the white boxed region.

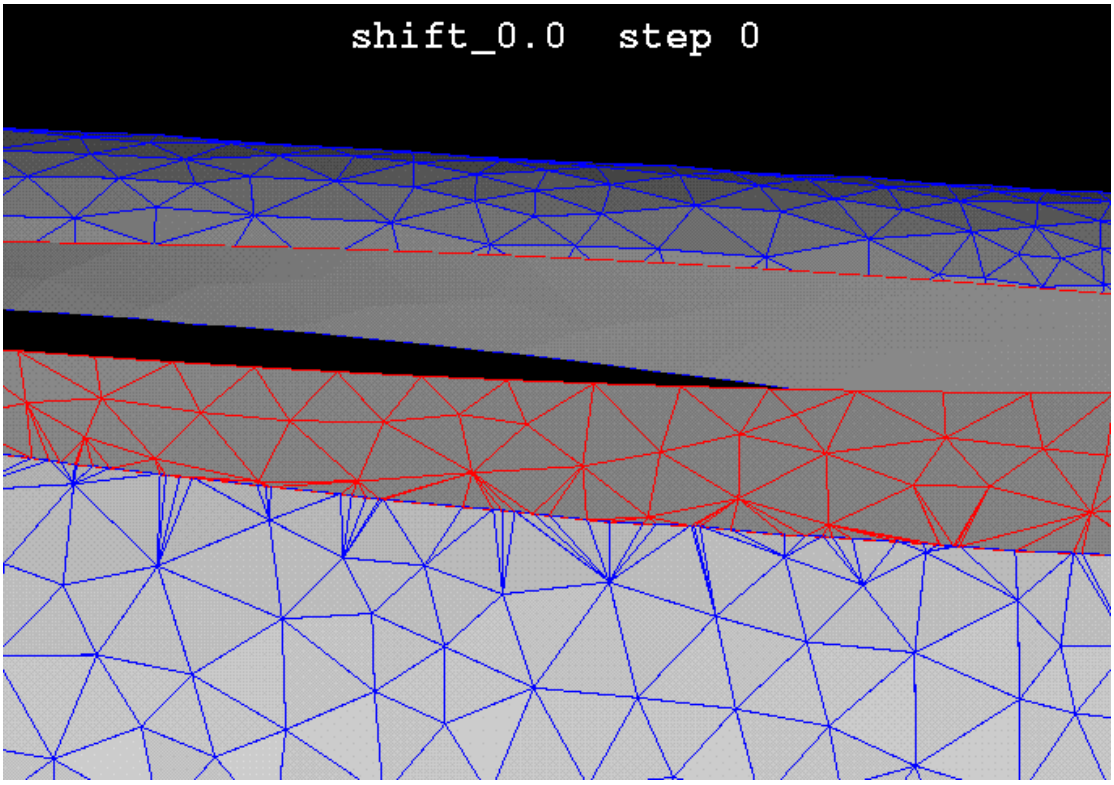


Figure 7(a). Intersection before ICU (step 0).

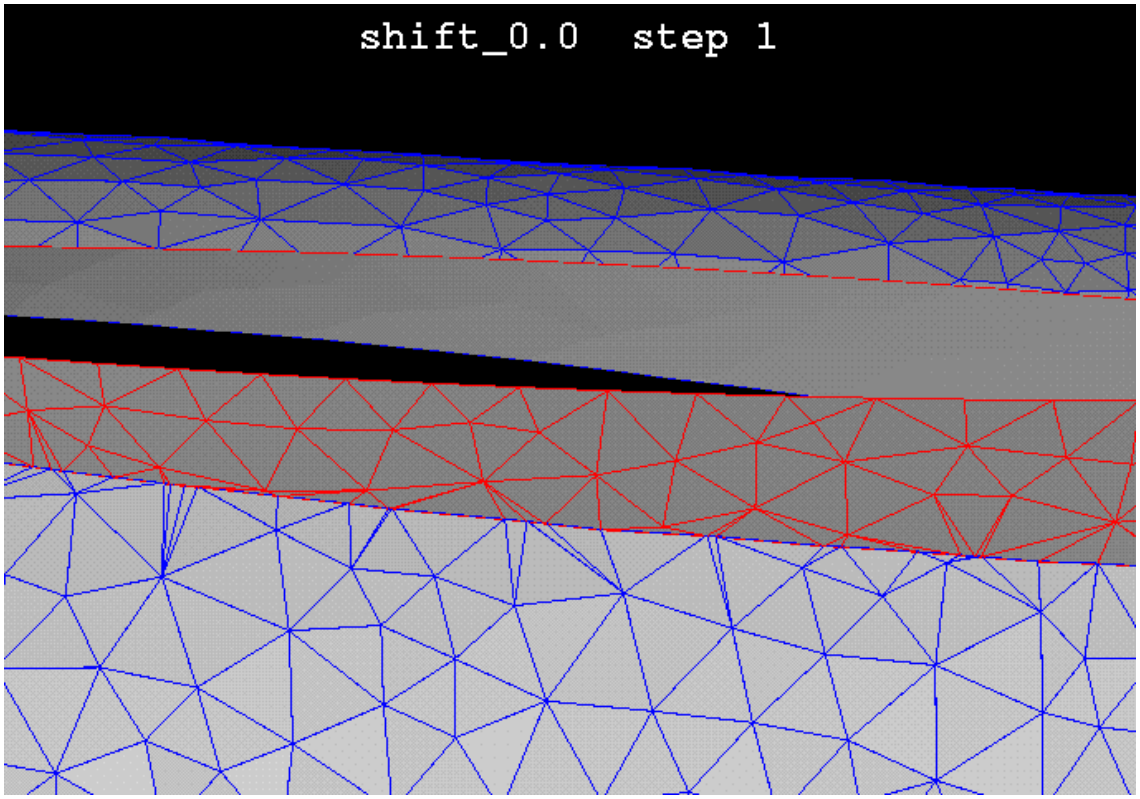
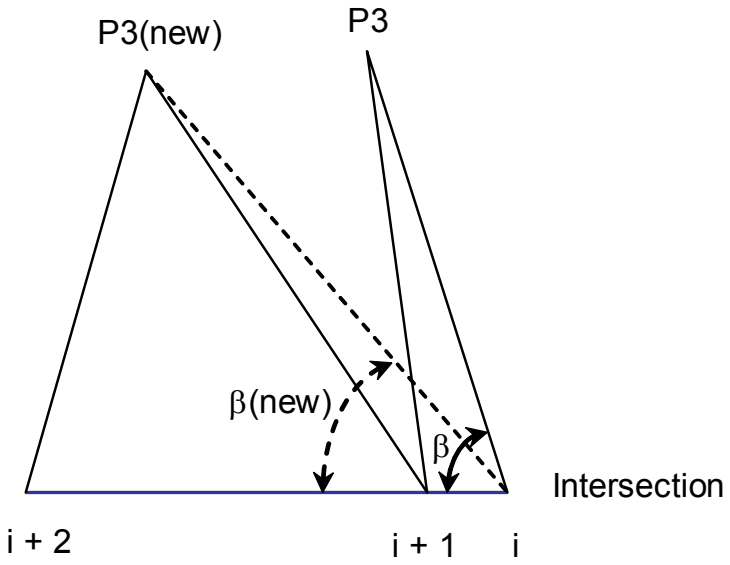
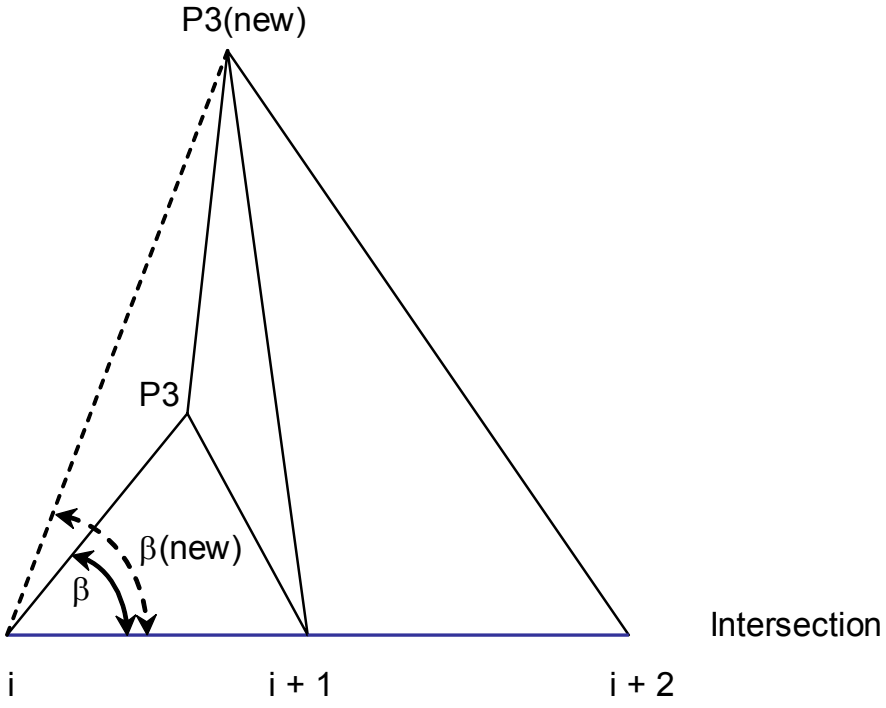


Figure 7(b). Intersection after step 1 of the ICU program.



No cross-over will occur, β is greater than β(new).

Figure 8(a). Example of an acceptable edge collapse from i + 1 to i (step 2).



Cross-over will occur, β is less than β(new).

Figure 8(b). Example of a nonacceptable edge collapse from i + 1 to i (step 2).

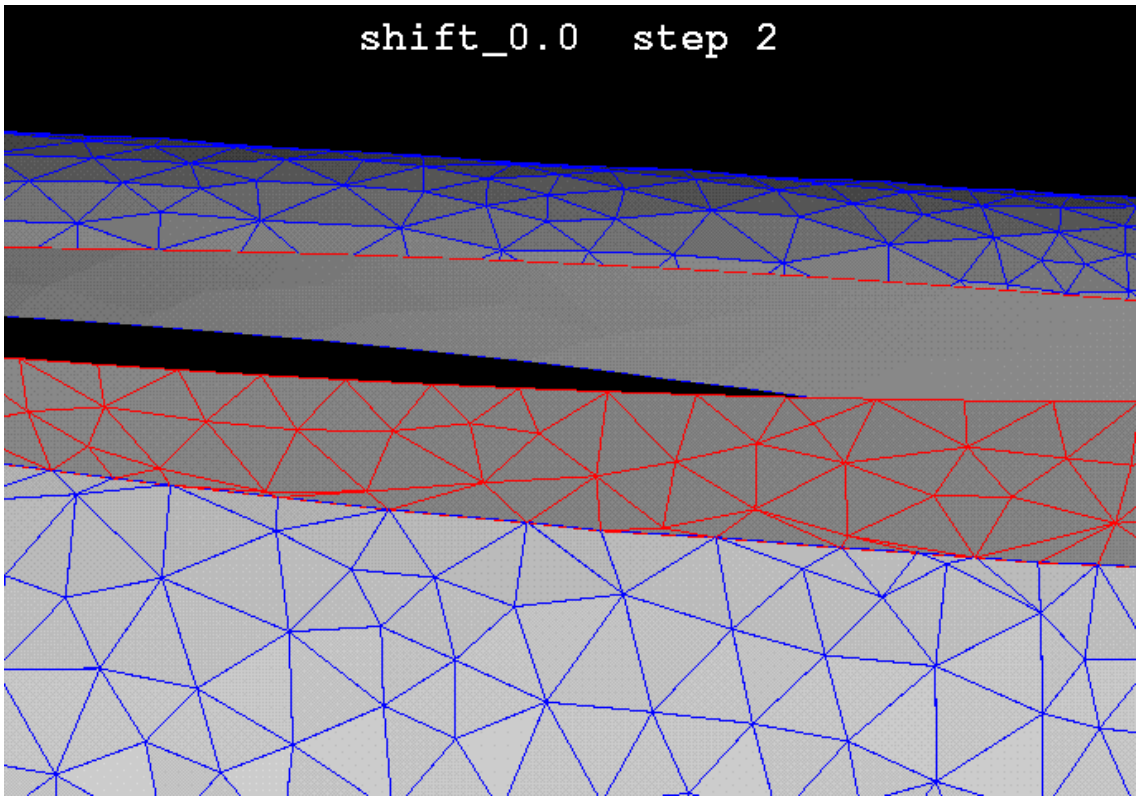


Figure 9. Intersection after step 2 of the ICU program.

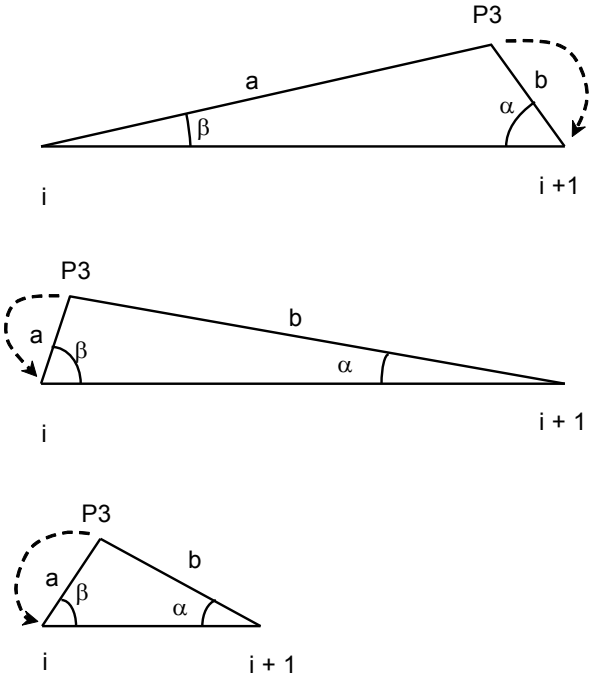


Figure 10. Step 3 examples of criteria for collapsing edges of a component to the intersection line; top shows if β is small, middle if α is small, and bottom if *a*norm or *b*norm are less than a specified amount of the normalized distribution curve.

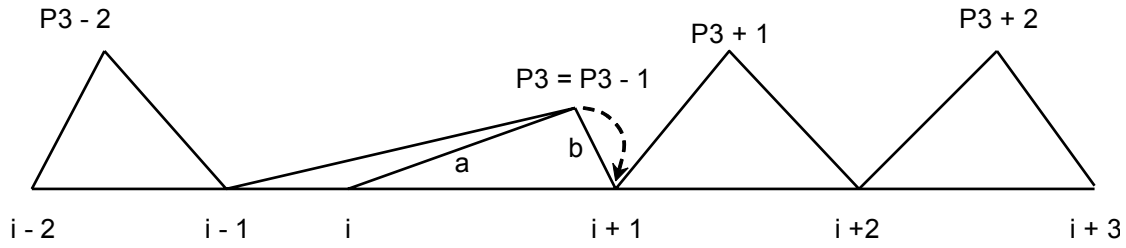


Figure 11. Example 1 of a noncollapsible triangle.

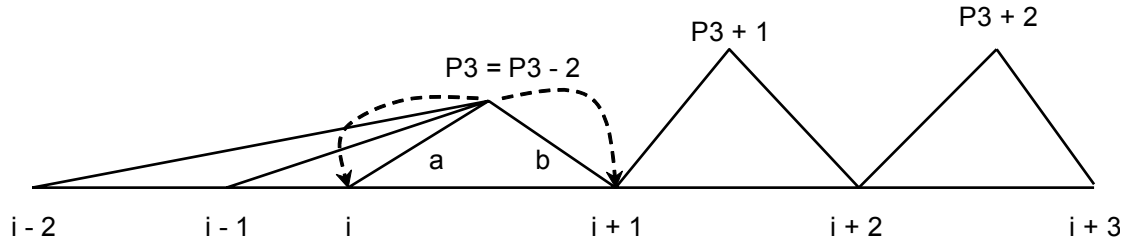


Figure 12. Example 2 of a noncollapsible triangle.

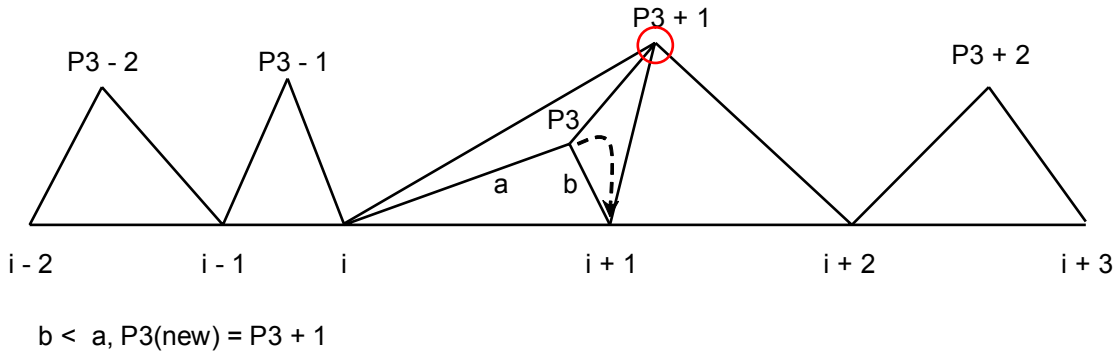


Figure 13. Example 3 of a situation where the red circled point is nonreplaceable after P3 is replaced with i + 1 (triangle with vertices i, i + 1, P3 is collapsed).

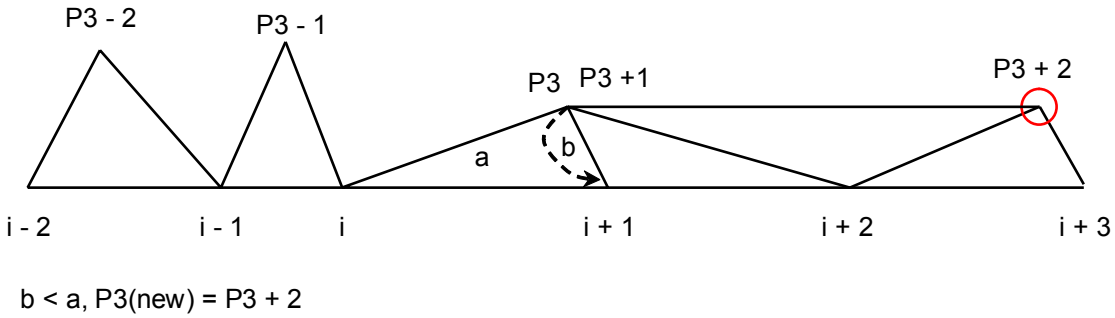


Figure 14. Example 4 of a situation where after the triangle with vertices i, i + 1, P3 is collapsed, the red circled point is nonreplaceable.

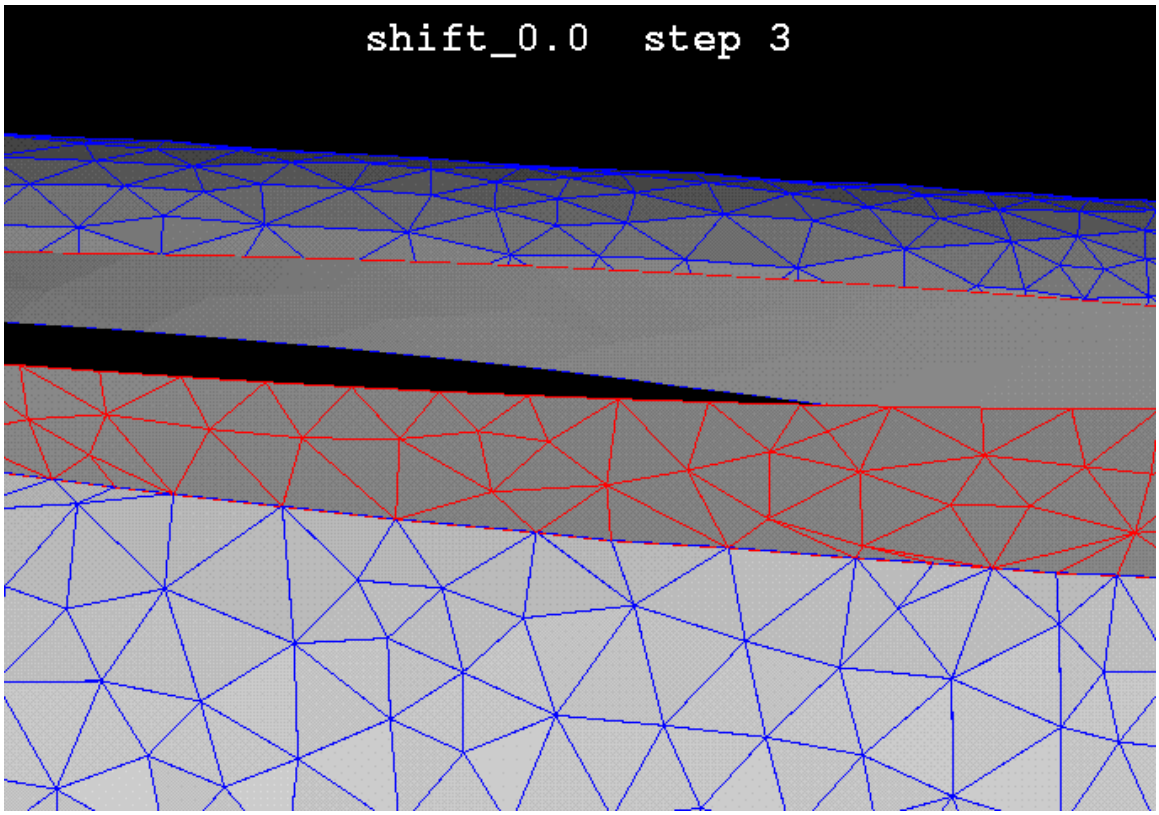


Figure 15. Intersection after step 3 of the ICU program.

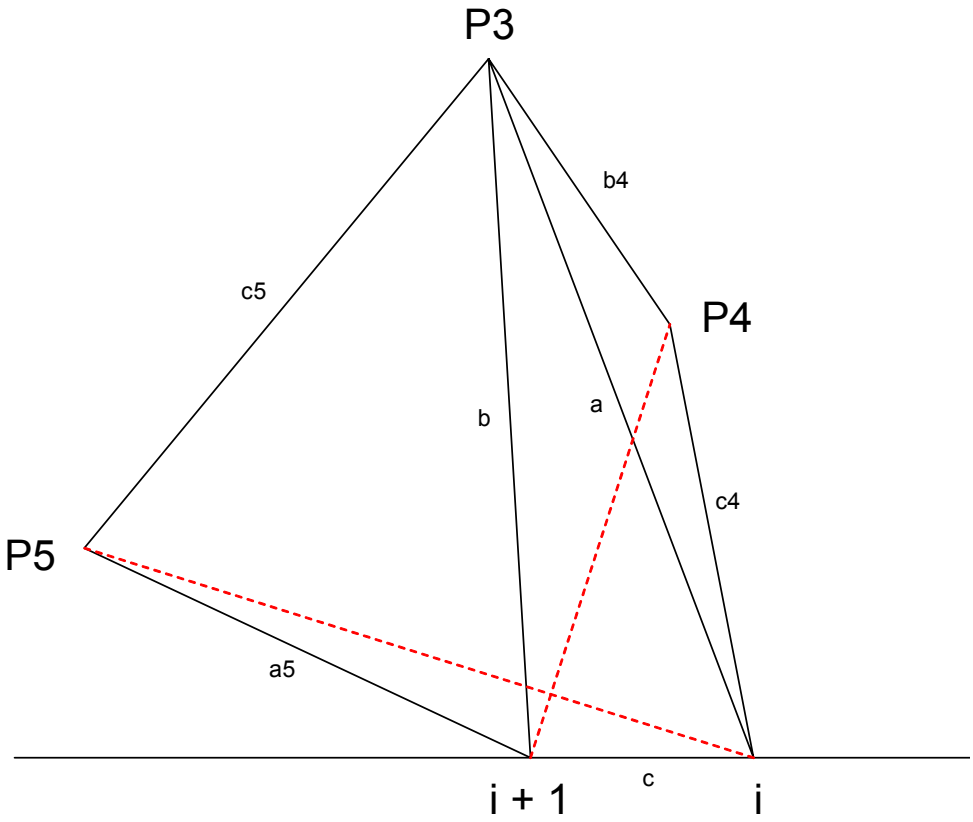


Figure 16. Diagram of P4- and P5-type edge swaps.

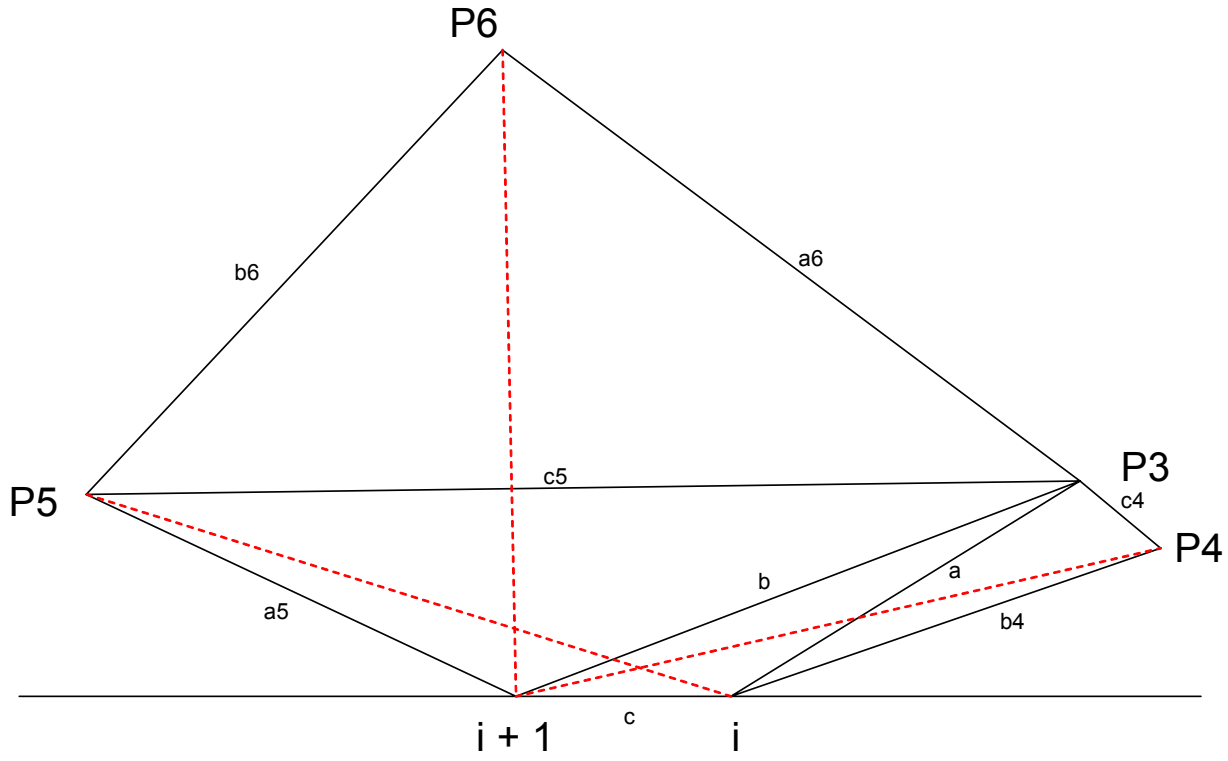


Figure 17. Diagram of a P6 swap.

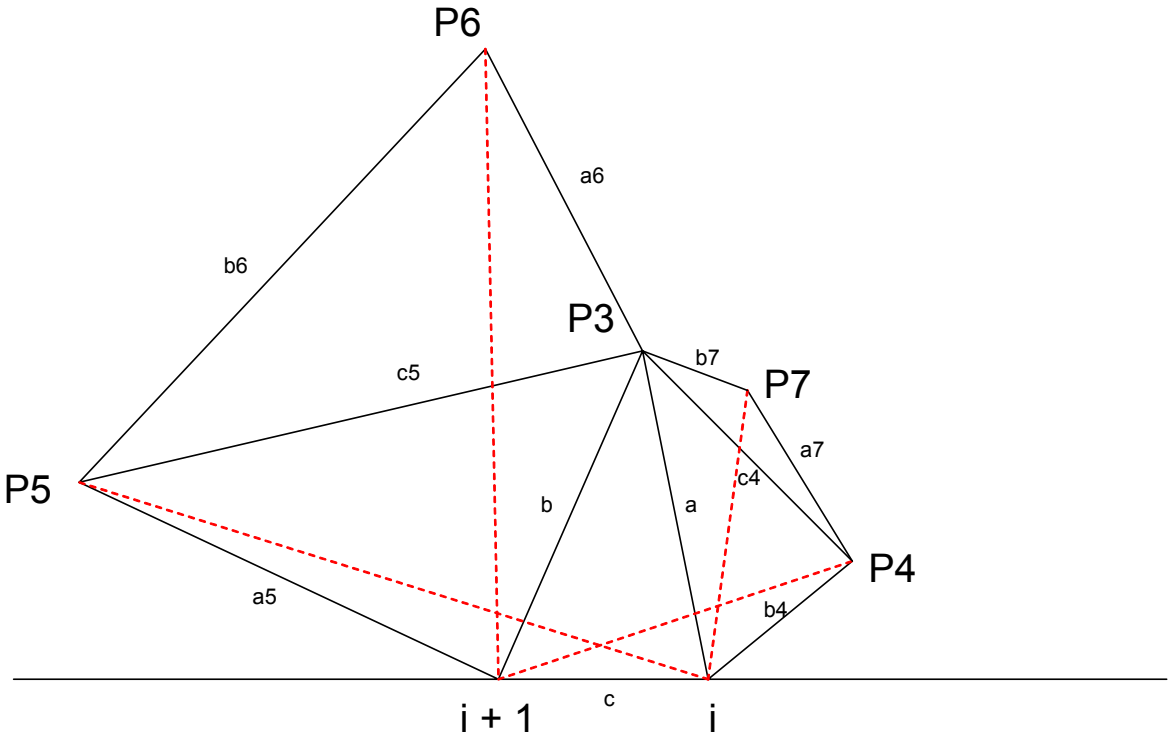


Figure 18. Diagram of a P7 swap.

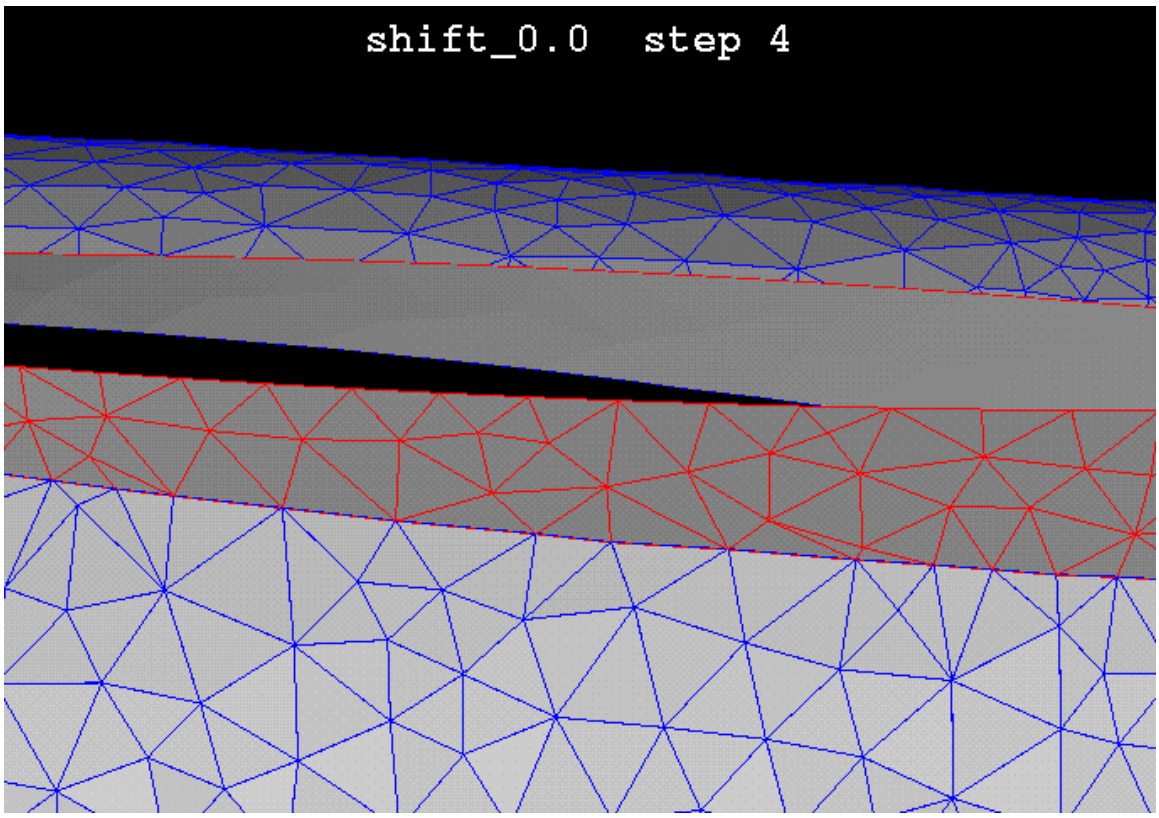


Figure 19. Intersection after step 4 of the ICU program.

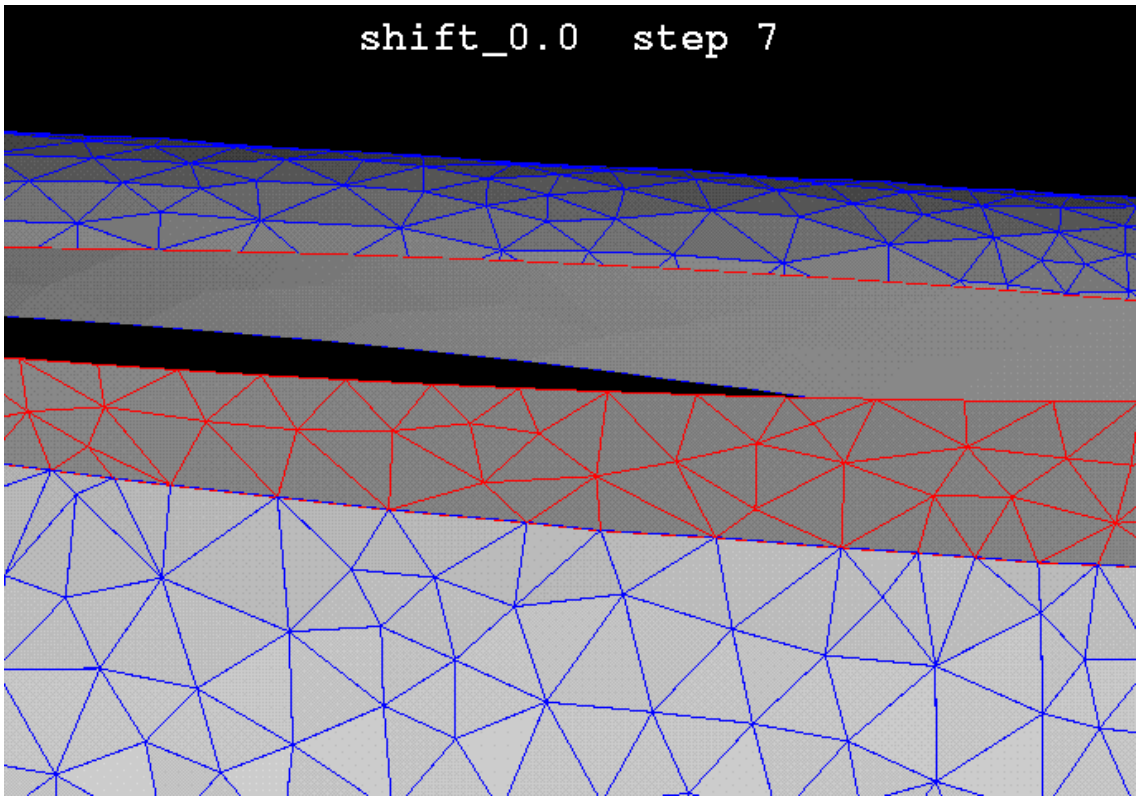


Figure 20. Intersection after step 7 of the ICU program.

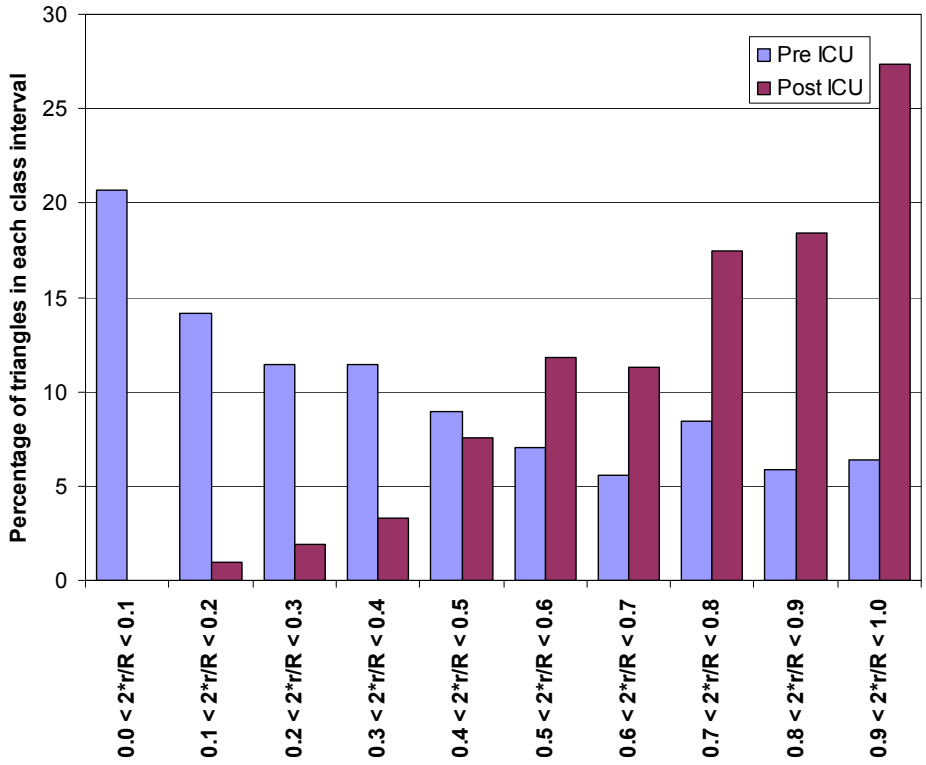


Figure 21. Quality assessment statistics of triangles along the intersection, before and after application of ICU with wing in original position (shift = 0.0).

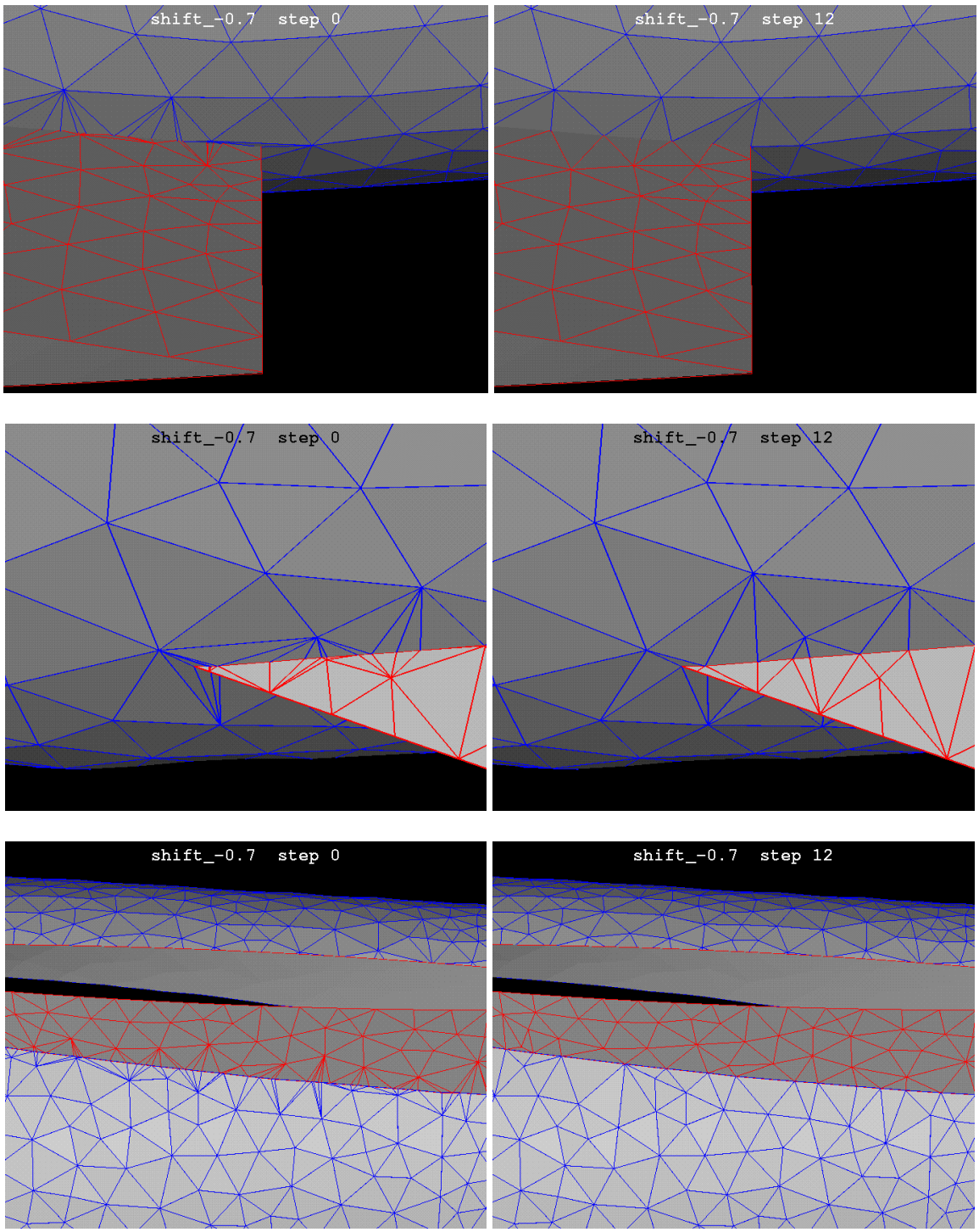


Figure 22. Example 1 of the qualitative assessments of the sharp-leading-edge wing test case before and after application of ICU. Wing shifted –0.7 inches before intersection. Top: trailing edge from above; mid: leading edge from above; lower: midchord of the wing from below.

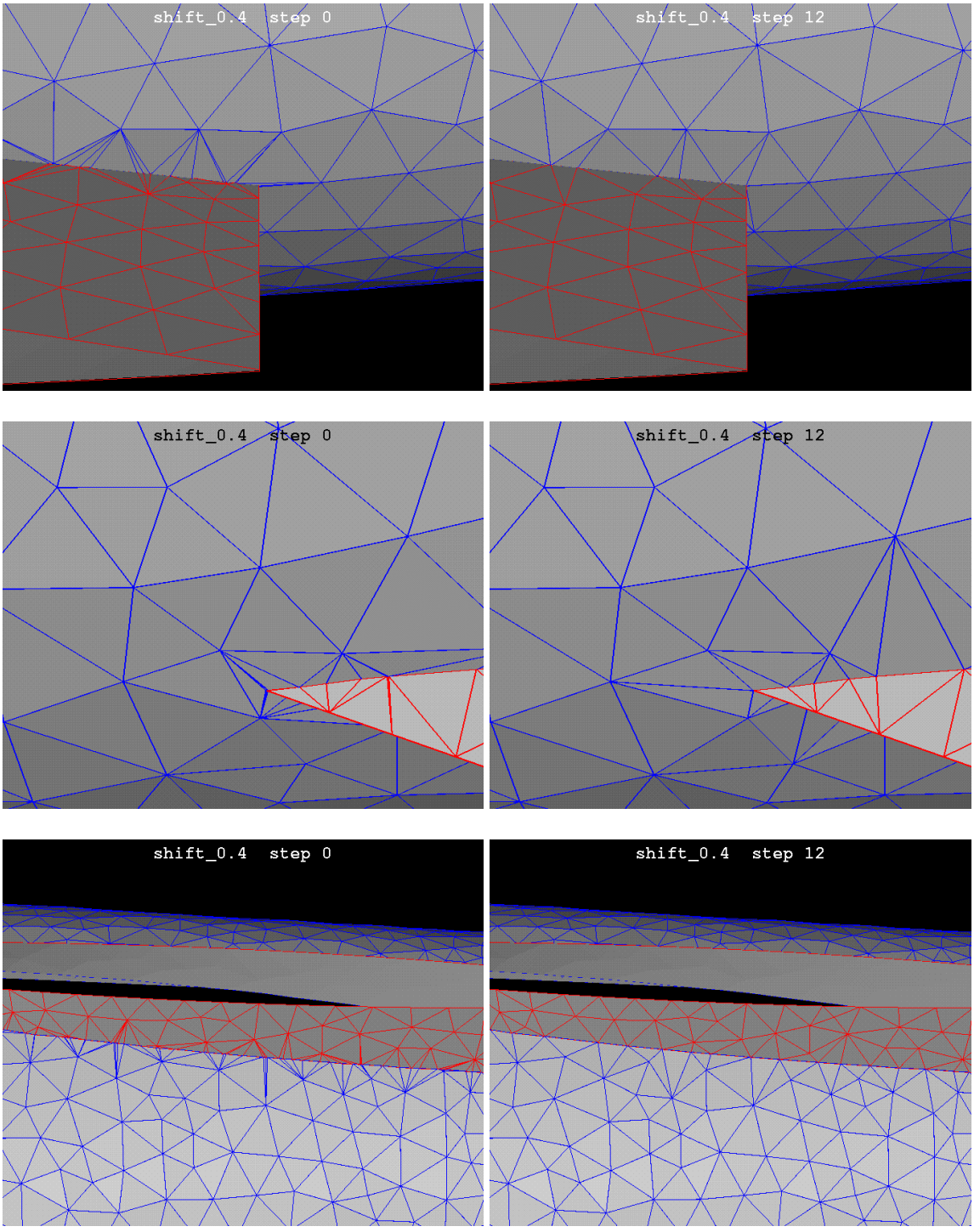


Figure 23. Example 2 of the qualitative assessments of the sharp-leading-edge wing test case before and after application of ICU. Wing shifted 0.4 inches before intersection. Top: trailing edge from above; mid: leading edge from above; lower: mid-chord of the wing from below.

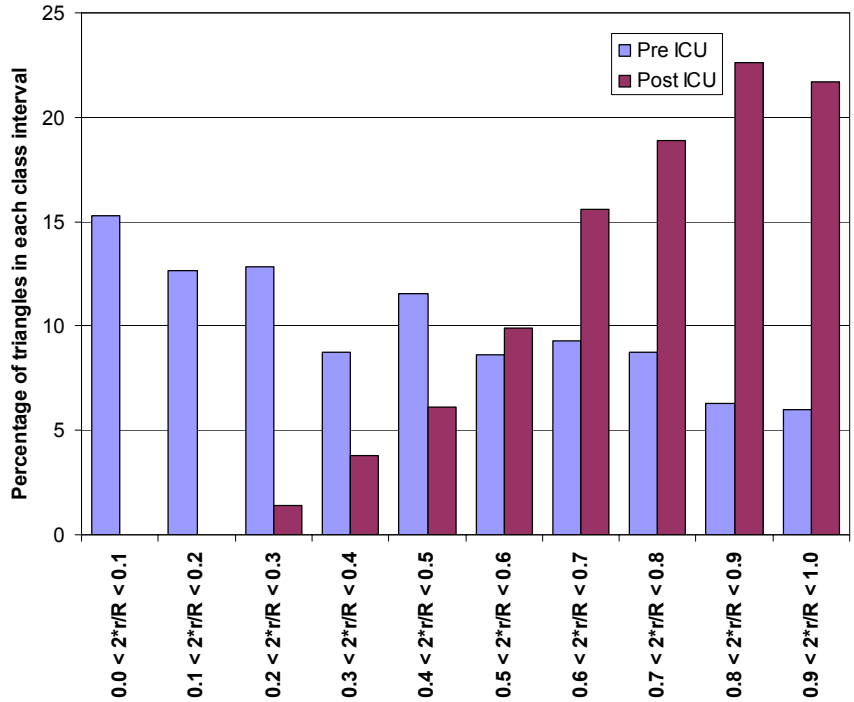


Figure 24. Quality statistics for example 1 (fig. 22) of the sharp-leading-edge wing test case with wing shifted –0.7 inches.

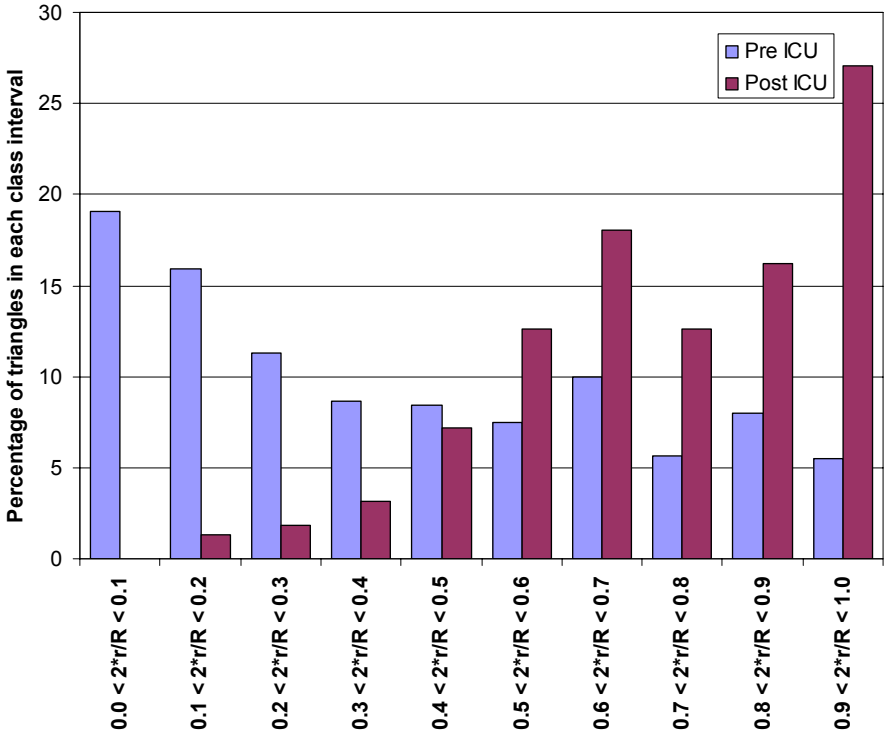


Figure 25. Quality statistics for example 2 (fig. 23) of the sharp-leading-edge wing test case with wing shifted 0.4 inches.

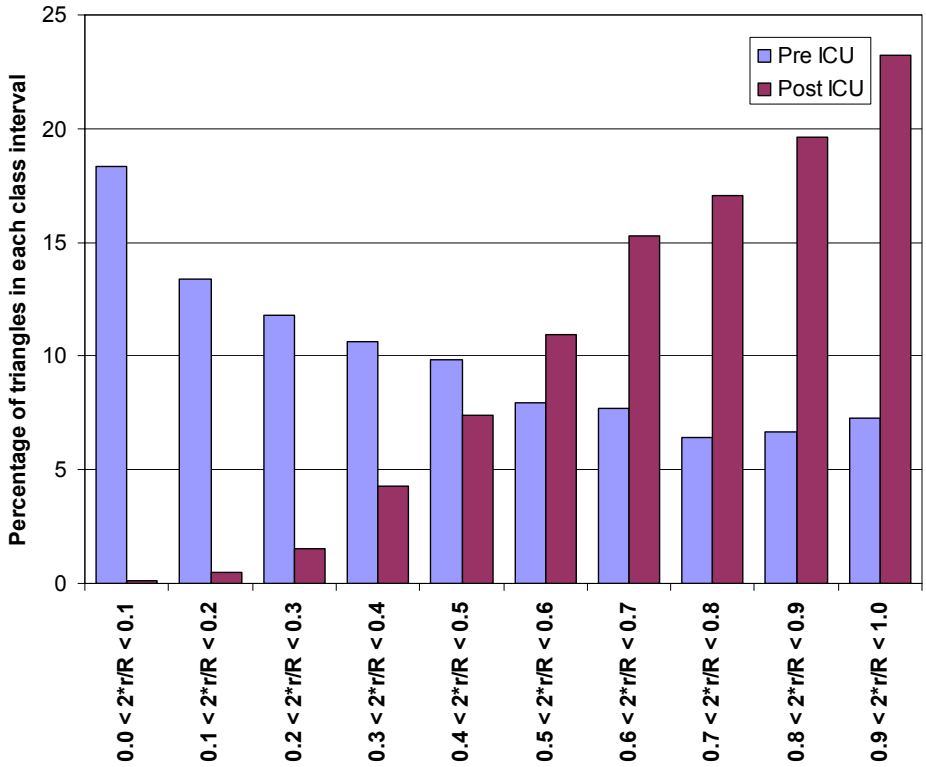


Figure 26. Quality statistics of the average for all 41 different wing positions of the sharp-leading-edge wing test cases.

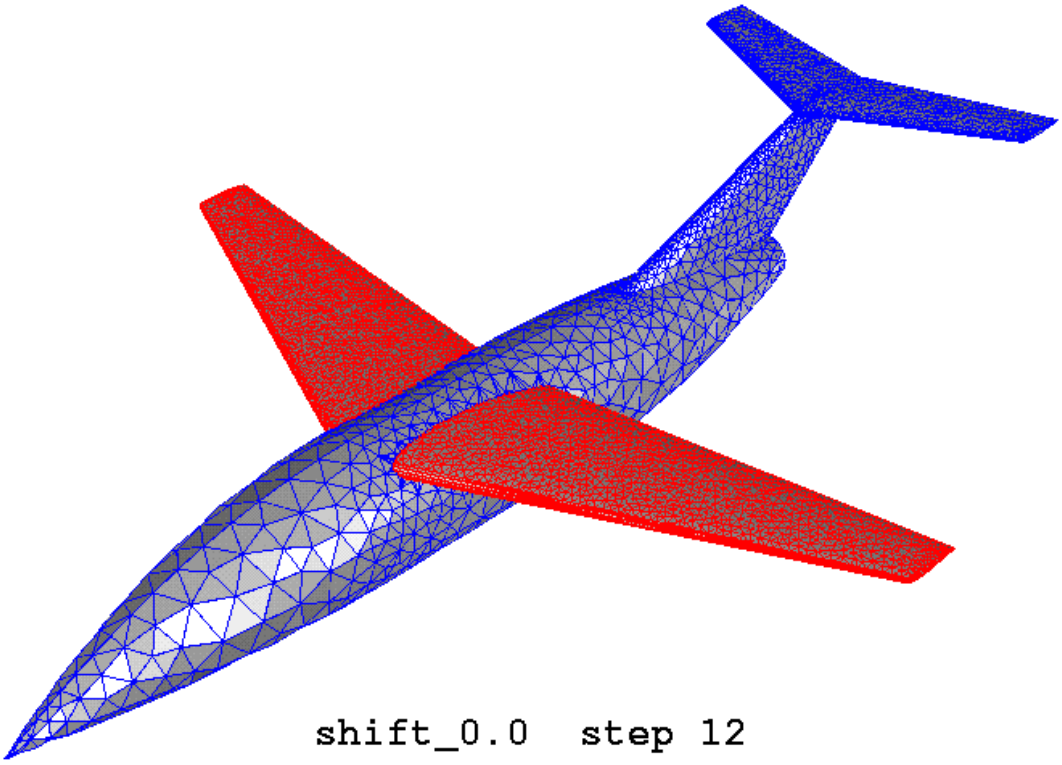


Figure 27(a). Blunt-leading-edge test case with wing in most upward position, coarse grid.

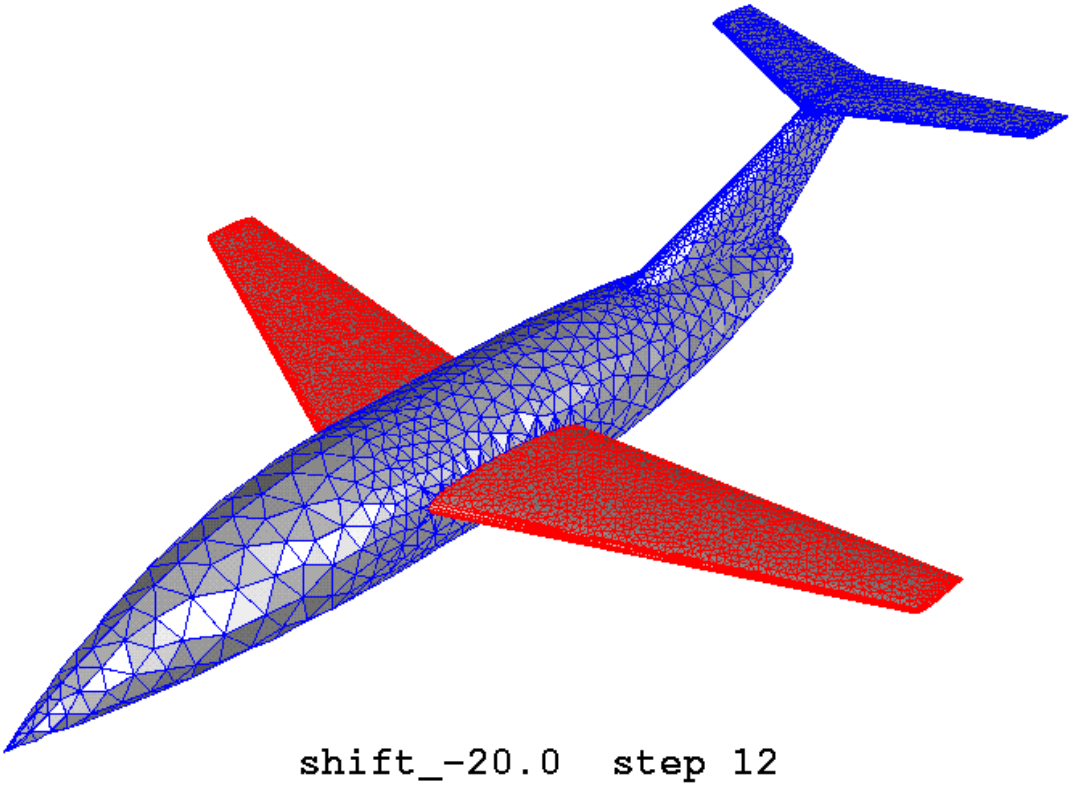


Figure 27(b). Blunt-leading-edge test case with wing in lowest test position, coarse grid.

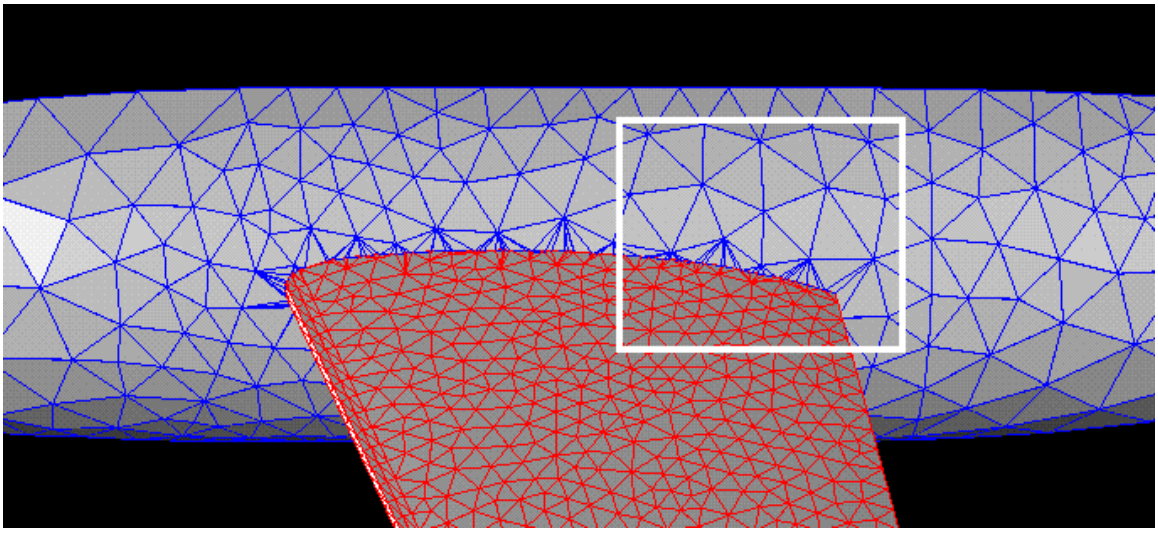


Figure 28(a). Trailing edge viewing region.

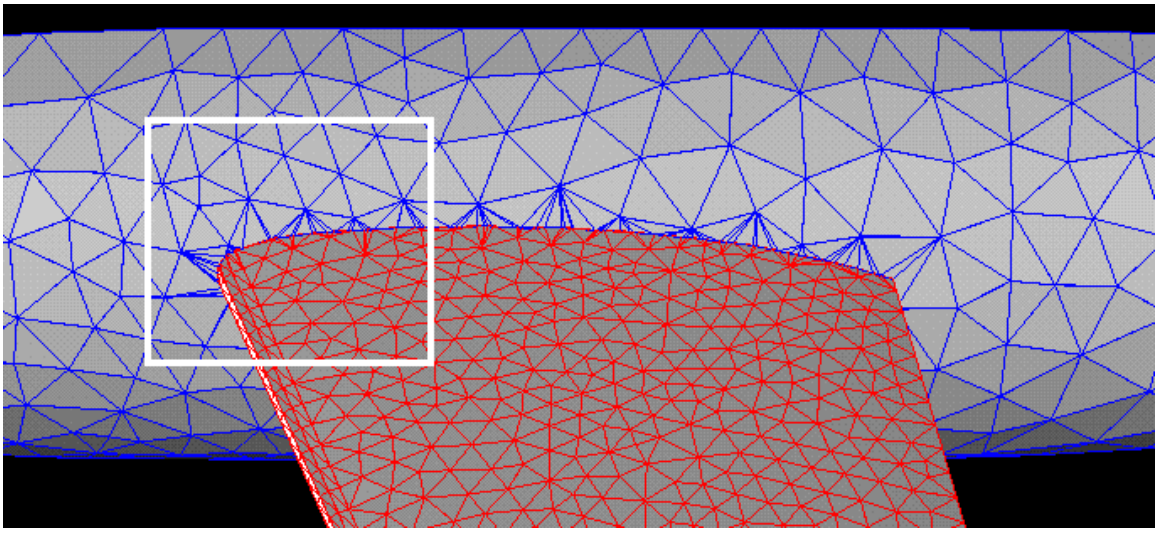


Figure 28(b). Leading edge viewing region.

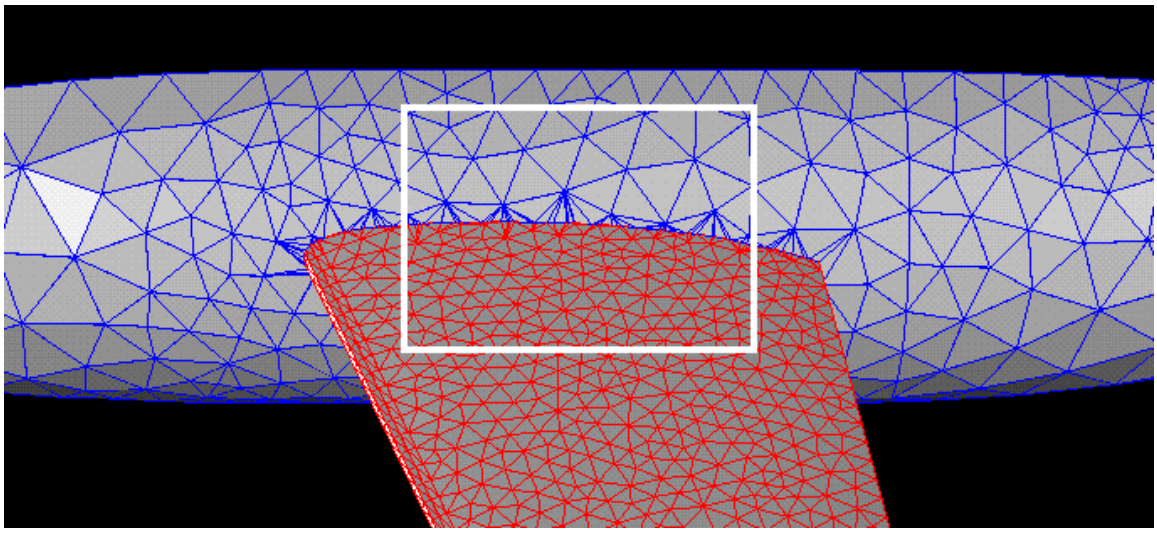


Figure 28(c). Midwing viewing region.

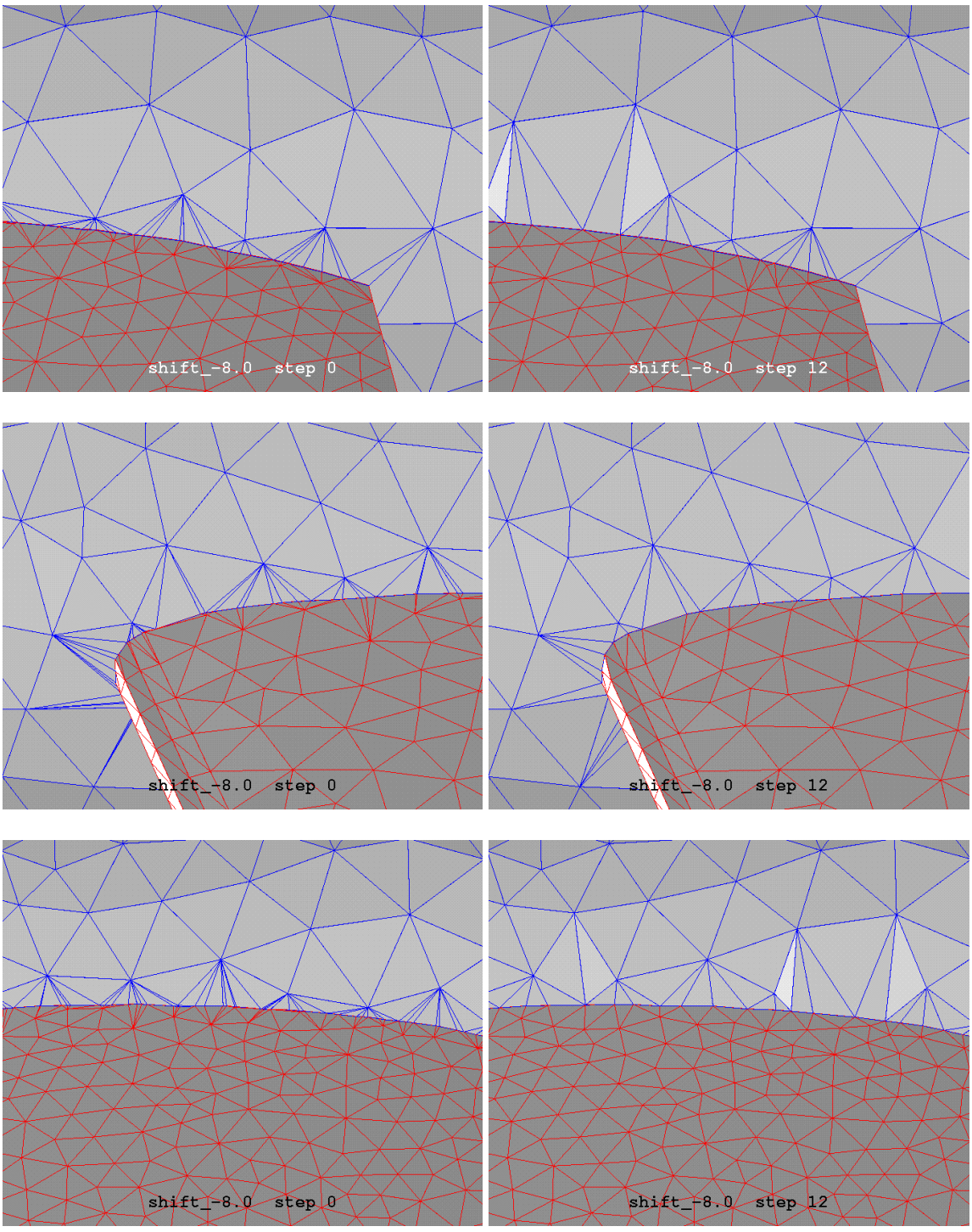


Figure 29. Refinement level 1 (coarse) of the blunt-leading-edge test case before and after application of ICU. Wing shifted –8.0 inches before intersection. Top: trailing edge from above; mid: leading edge from above; lower: midchord of the wing from above.

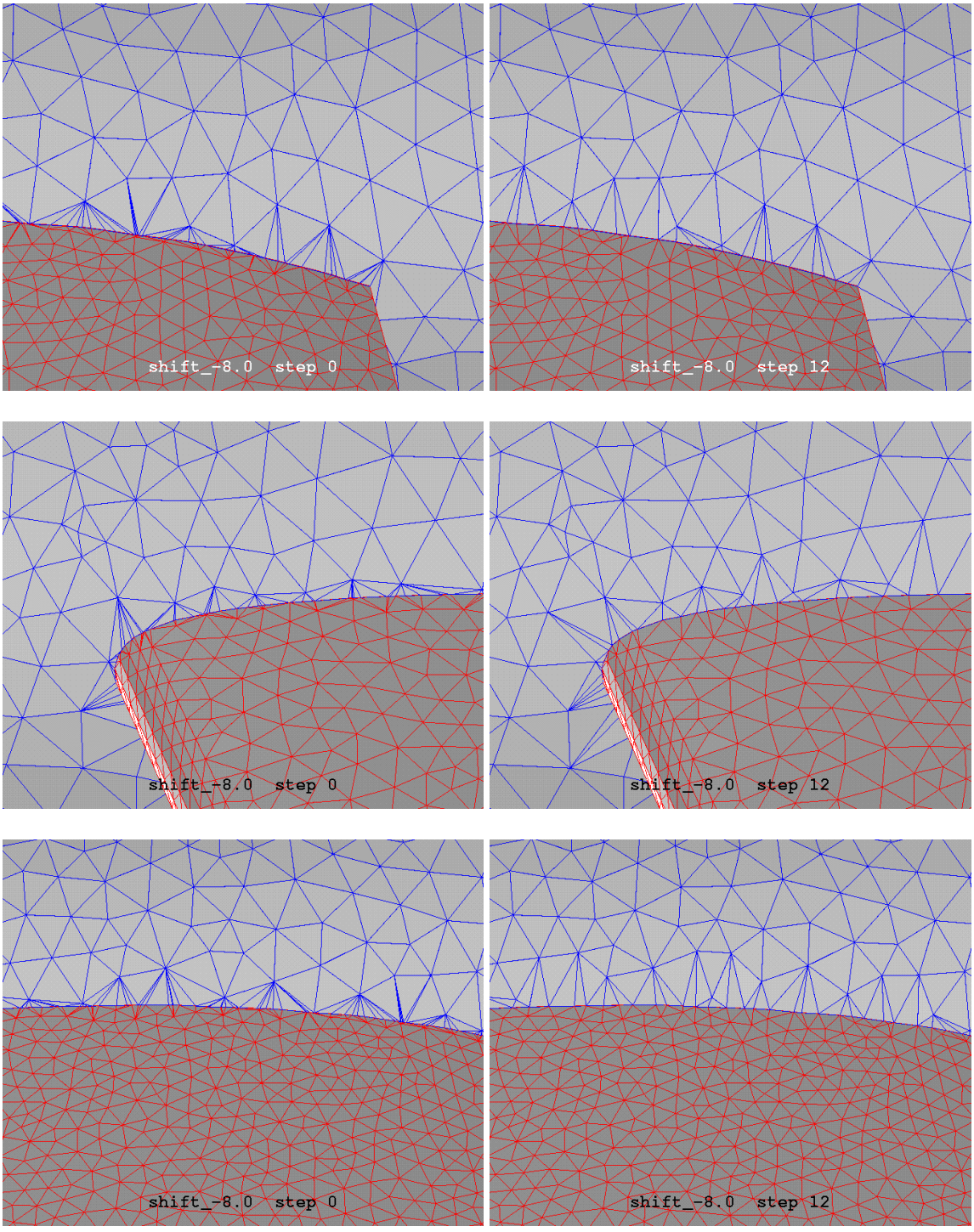


Figure 30. Refinement level 2 of the blunt-leading-edge test case before and after application of ICU. Wing shifted –8.0 inches before intersection. Top: trailing edge from above; mid: leading edge from above; lower: midchord of the wing from above.

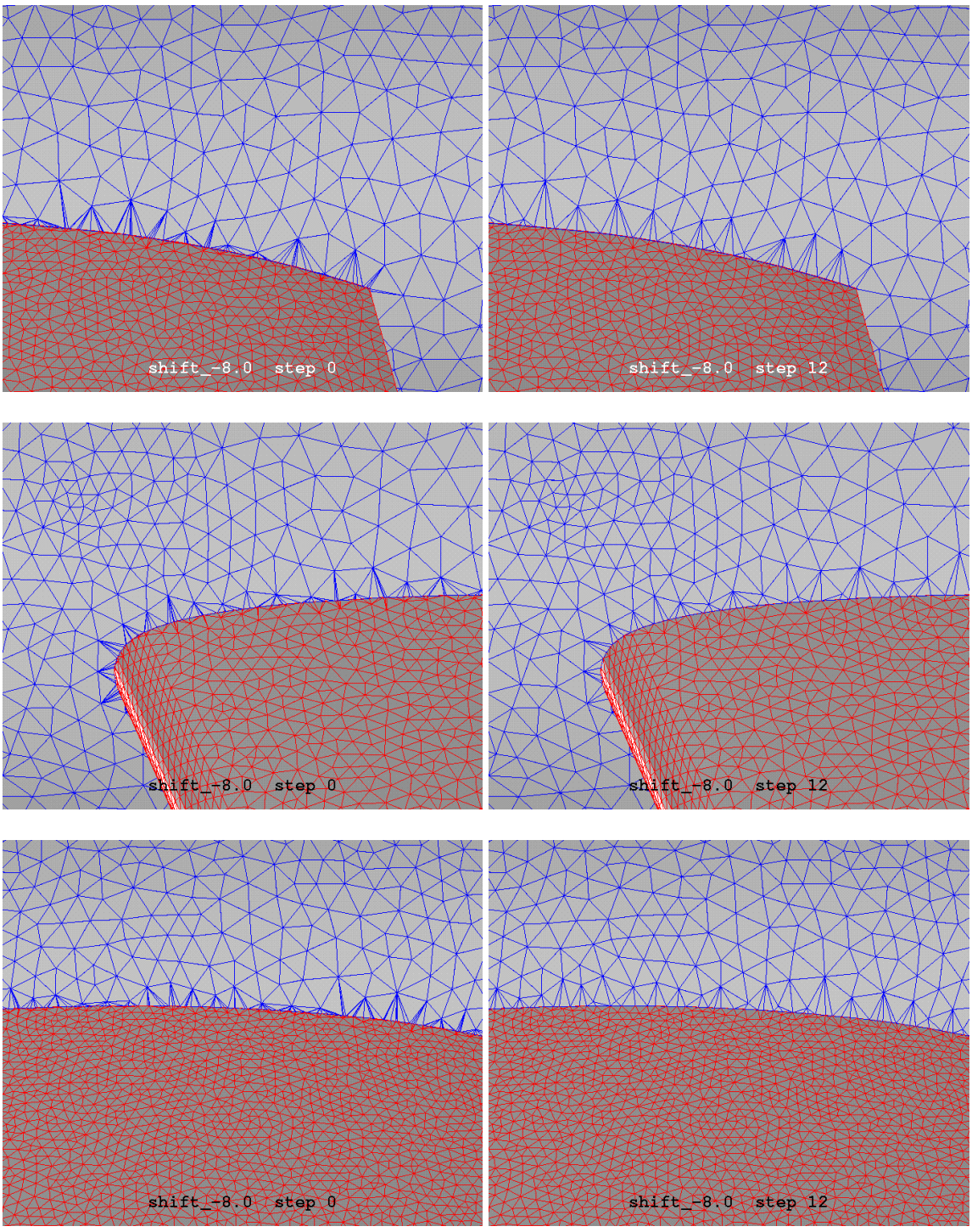


Figure 31. Refinement level 3 of the blunt-leading-edge test case before and after application of ICU. Wing shifted –8.0 inches before intersection. Top: trailing edge from above; mid: leading edge from above; lower: midchord of the wing from above.

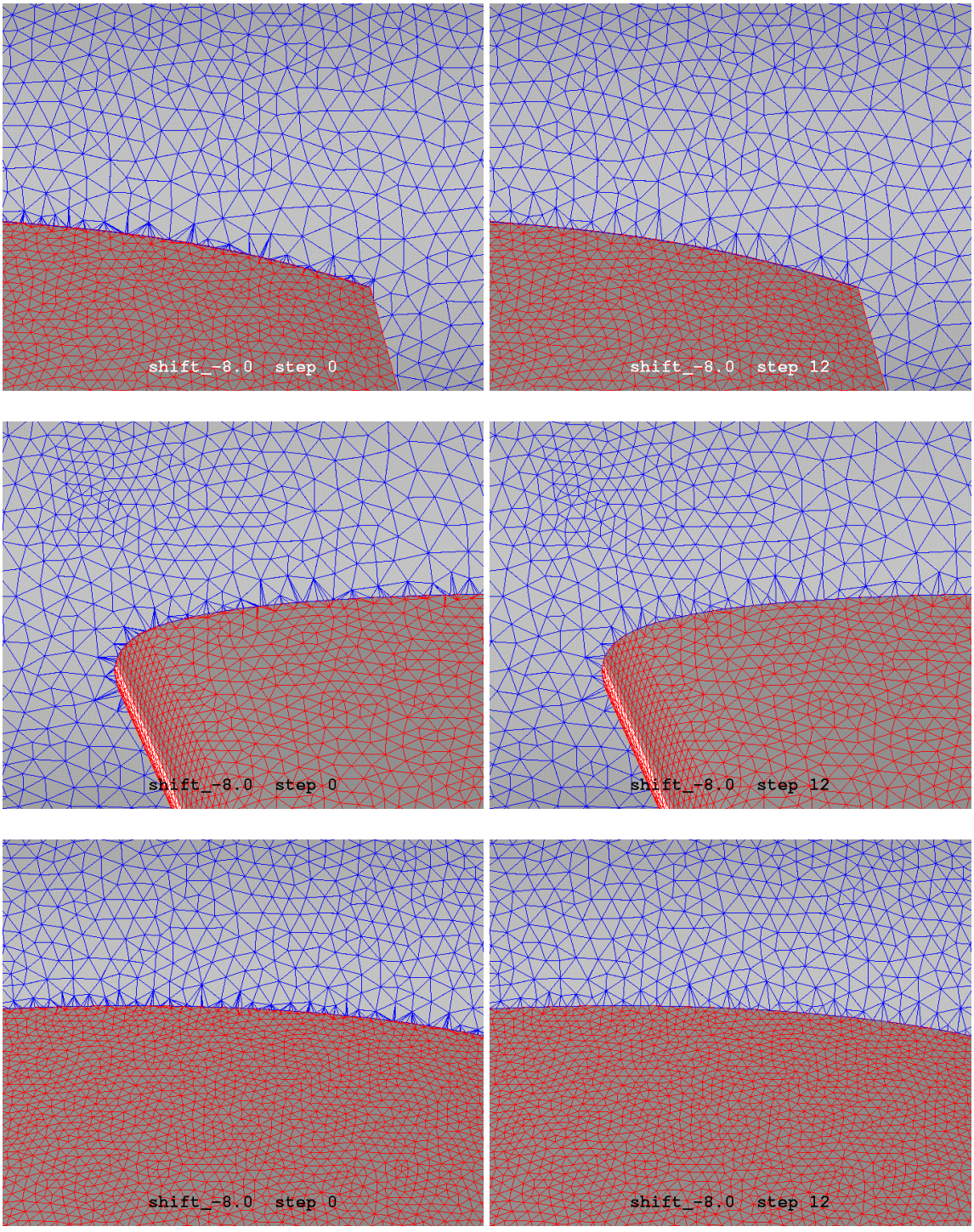


Figure 32. Refinement level 4 (fine) of the blunt-leading-edge test case before and after application of ICU. Wing shifted –8.0 inches before intersection. Top: trailing edge from above; mid: leading edge from above; lower: midchord of the wing from above.

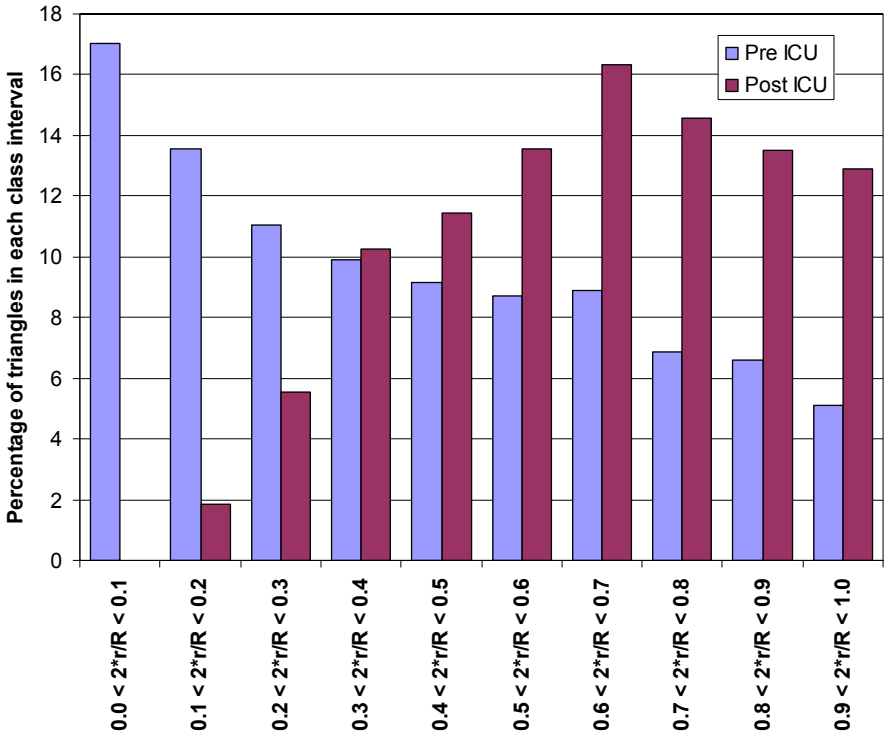


Figure 33. Quality statistics of the average for all 21 different wing positions of the blunt-leading-edge wing test cases with refinement level 1 meshes (coarse).

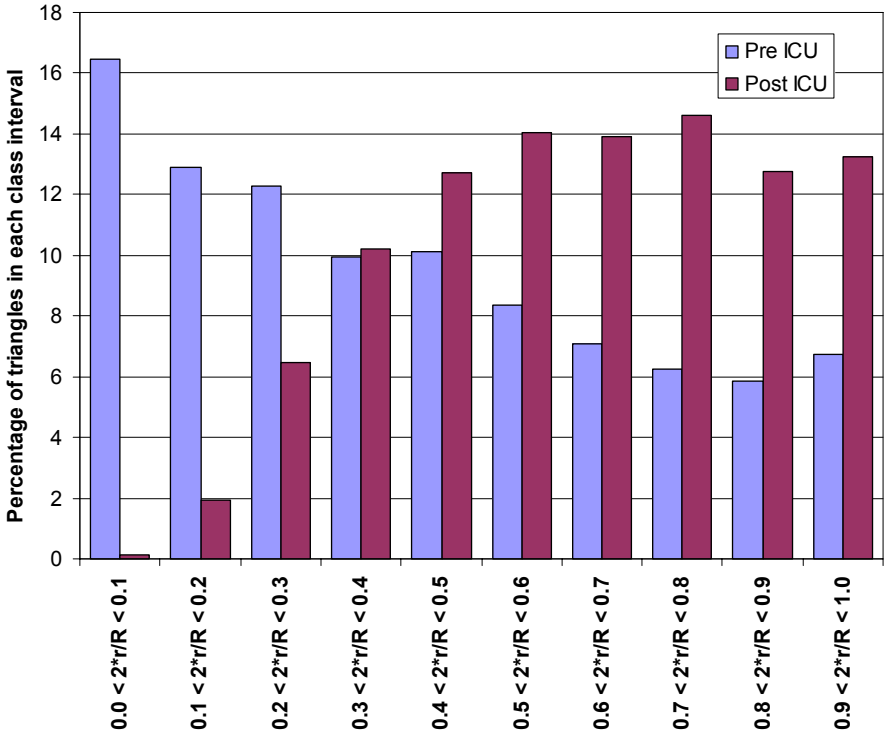


Figure 34. Quality statistics of the average for all 21 different wing positions of the blunt-leading-edge wing test cases with refinement level 2 meshes.

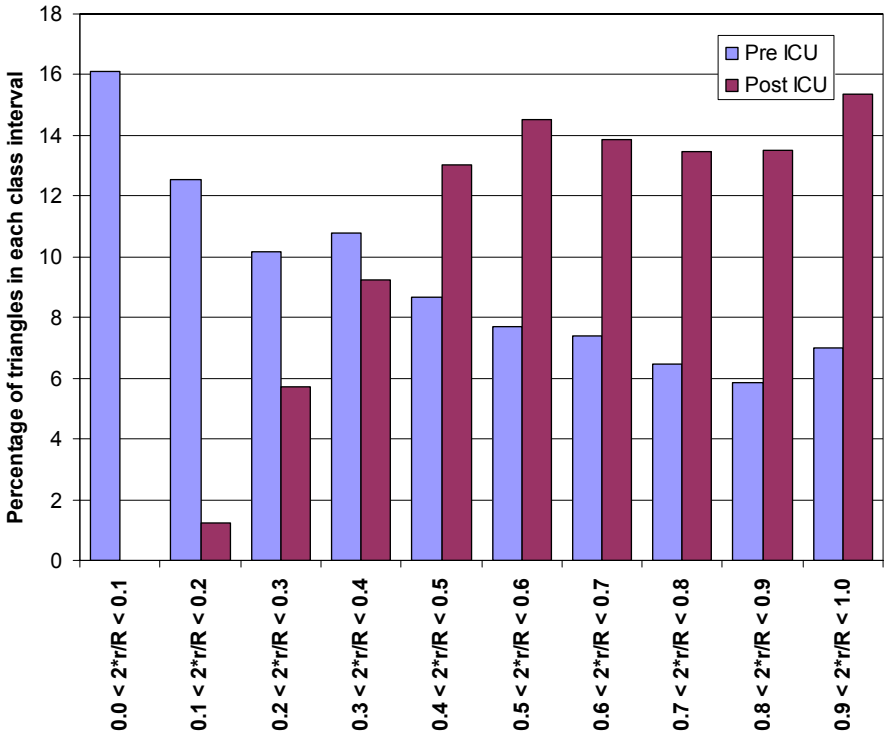


Figure 35. Quality statistics of the average for all 21 different wing positions of the blunt-leading-edge wing test cases with refinement level 3 meshes.

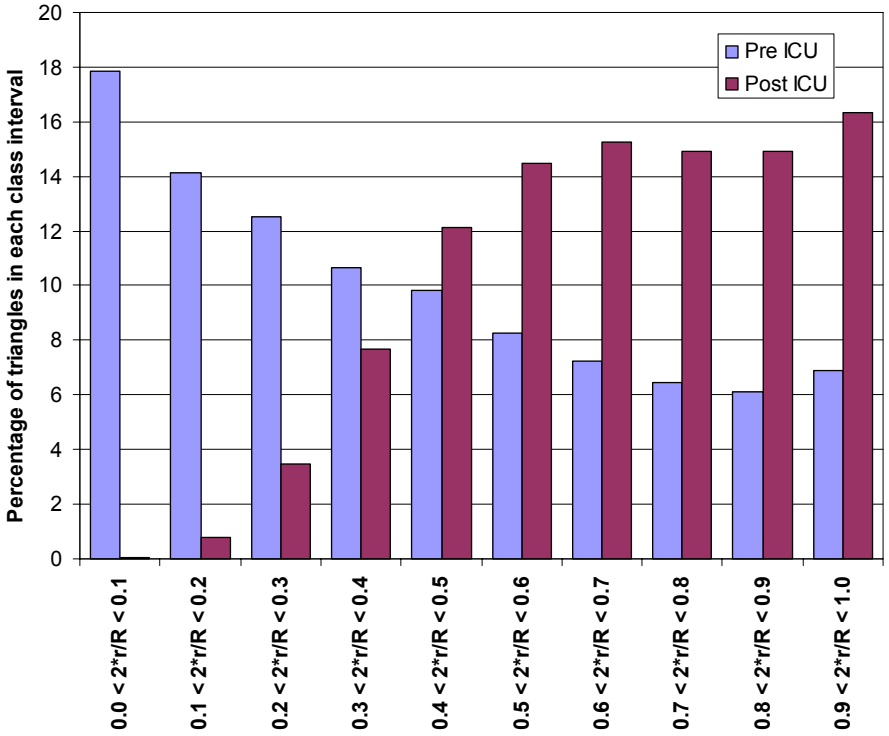


Figure 36. Quality statistics of the average for all 21 different wing positions of the blunt-leading-edge wing test cases with refinement level 4 meshes (fine).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
05/20/2005	Technical Memorandum	

4. TITLE AND SUBTITLE
A Program to Improve the Triangulated Surface Mesh Quality Along Aircraft Component Intersections

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Susan E. Cliff

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
21-982-10-20

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Ames Research Center
Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER
A-0513777

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546-0001

10. SPONSORING/MONITOR'S ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/TM-2005-213455

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified — Unlimited Distribution: Nonstandard
Subject Category: 05, 61
Availability: NASA CASI (301) 621-0390

13. SUPPLEMENTARY NOTES
POC: Susan E. Cliff, Ames Research Center, Analysis Branch, MS 258-1, Moffett Field, CA 94035-1000
(650) 604-3907

14. ABSTRACT
A computer program has been developed for improving the quality of unstructured triangulated surface meshes in the vicinity of component intersections. The method relies solely on point removal and edge swapping for improving the triangulations. It can be applied to any lifting surface component such as a wing, canard or horizontal tail component intersected with a fuselage, or it can be applied to a pylon that is intersected with a wing, fuselage or nacelle. The lifting surfaces or pylon are assumed to be aligned in the axial direction with closed trailing edges. The method currently maintains salient edges only at leading and trailing edges of the wing or pylon component. This method should work well for any shape of fuselage that is free of salient edges at the intersection. The method has been successfully demonstrated on a total of 125 different test cases that include both blunt and sharp wing leading edges. The code is targeted for use in the automated environment of numerical optimization where geometric perturbations to individual components can be critical to the aerodynamic performance of a vehicle. Histograms of triangle aspect ratios are reported to assess the quality of the triangles attached to the intersection curves before and after application of the program. Large improvements to the quality of the triangulations were obtained for the 125 test cases; the quality was sufficient for use with an automated tetrahedral mesh generation program that is used as part of an aerodynamic shape optimization method.

15. SUBJECT TERMS
Mesh quality, Optimization, Triangulation, Intersection, Cell quality, Tetrahedral meshes

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Susan E. Cliff
Unclassified	Unclassified	Unclassified	Unclassified	42	19b. TELEPHONE *(Include area code)* (650) 604-3907